— APPLESOFT —

5/26/82

```
*********************************
*                               *
*          Applesoft            *
*                               *
* Applesoft is a copyrighted    *
* program of Apple Computer,    *
* Inc. and Microsoft, Inc.      *
*                               *
* This source listing is for    *
* use only by the rightful      *
* owner of the copy of          *
* Applesoft from which it was   *
* generated and may not be      *
* published or distributed to   *
* others, whether or not for    *
* commercial purposes.          *
*                               *
*********************************
*                               *
*  Put SYM $8000 to assemble.   *
*                               *
*********************************
```

From the archives of
Roger Wagner Publishing, Inc.

APPLE II FOREVER
kansasfest

```
1     **********************************
2     *                                *
3     *          Applesoft             *
4     *                                *
5     * Applesoft is a copyrighted     *
6     * program of Apple Computer,     *
7     * Inc. and Microsoft, Inc.       *
8     *                                *
9     * This source listing is for     *
10    * use only by the rightful       *
11    * owner of the copy of           *
12    * Applesoft from which it was    *
13    * generated and may not be       *
14    * published or distributed to    *
15    * others, whether or not for     *
16    * commercial purposes.           *
17    *                                *
18    **********************************
19    *                                *
20    *  Put SYM $8000 to assemble.    *
21    *                                *
22    **********************************
23
24              ERR   ($75)        ;Check that SYM is set
25              ERR   ($76)-$80    ; to $8000
26
27    SAVEOBJ   KBD
28
29
30
31              ORG   $D000
32
33              DO    0
34
35    INCR      MAC
36              INC   ]1
37              BNE   NI
38              INC   ]1+1
39    NI        <<<
40
41    DECR      MAC
42              LDX   ]1
43              BNE   ND
44              DEC   ]1+1
45    ND        DEC   ]1
46              <<<
47
48    CMPR      MAC
49              LDA   ]1
50              CMP   ]2
51              LDA   ]1+1
52              SBC   ]2+1
53              <<<
54
55    TR        MAC
56              LDA   ]1
57              STA   ]2
58              <<<
59
```

```
60    TRDB      MAC
61              >>>   TR.]1;]2
62              >>>   TR.]1+1;]2+1
63              <<<
64
65    TRX       MAC
66              LDX   ]1
67              STX   ]2
68              LDX   ]1+1
69              STX   ]2+1
70              <<<
71
72    TRAX      MAC
73              LDA   ]1
74              LDX   ]1+1
75              STA   ]2
76              STX   ]2+1
77              <<<
78
79    TRAY      MAC
80              LDA   ]1
81              LDY   ]1+1
82              STA   ]2
83              STY   ]2+1
84              <<<
85
86    TRXY      MAC
87              LDX   ]1
88              LDY   ]1+1
89              STX   ]2
90              STY   ]2+1
91              <<<
92
93    AD        MAC
94              LDA   ]1
95              ADC   ]2
96              STA   ]3
97              <<<
98
99    ADD       MAC
100             CLC
101             >>>   AD.]1;]2;]3
102             >>>   AD.]1+1;]2+1;]3+1
103             <<<
104
105   SB        MAC
106             LDA   ]1
107             SBC   ]2
108             STA   ]3
109             <<<
110
111   SUB       MAC
112             SEC
113             >>>   SB.]1;]2;]3
114             >>>   SB.]1+1;]2+1;]3+1
115             <<<
116
```

```
 117  BUMP      MAC
 118            ADC  ]1
 119            STA  ]1
 120            BCC  BPD
 121            INC  ]1+1
 122  BPD       <<<
 123
 124  PUSH      MAC
 125            LDA  ]1+1
 126            PHA
 127            LDA  ]1
 128            PHA
 129            <<<
 130
 131  PULL      MAC
 132            PLA
 133            STA  ]1
 134            PLA
 135            STA  ]1+1
 136            <<<
 137
 138  NEG       MAC
 139            LDA  ]1
 140            EOR  #$FF
 141            STA  ]1
 142            <<<
 143
 144            FIN
 145
 146            DO   SAVEOBJ
 147            DSK  APPLESOFT.OBJ
 148            FIN
 149
 150            PUT  APSOFT I
>1     ********************************
>2     *                              *
>3     *    Applesoft - Part I        *
>4     *                              *
>5     * Copywrite Apple Computer,    *
>6     * Inc. and Microsoft, Inc.;    *
>7     * not for publication or       *
>8     * distribution.                *
>9     *                              *
>10    ********************************
>11    *                              *
>12    *   Input parsing,             *
>13    *   Routine addressing,        *
>14    *   For-next loops, etc.       *
>15    *                              *
>16    *       $D000 - $DD7A          *
>17    *                              *
>18    ********************************
>19
>20    * Equates for all parts:
>21
>22    * Applesoft tokens:
>23
```

```
>24    for       =    $81
>25    data      =    $83
>26    pop       =    $A1
>27    goto      =    $AB
>28    gosub     =    $B0
>29    rem       =    $B2
>30    print     =    $BA
>31    tab       =    $C0
>32    to        =    $C1
>33    fn        =    $C2
>34    spc       =    $C3
>35    then      =    $C4
>36    at        =    $C5
>37    not       =    $C6
>38    step      =    $C7
>39    plus      =    $C8
>40    minus     =    $C9
>41    equal     =    $D0
>42    sgn       =    $D2
>43    scrn      =    $D7
>44    leftstr   =    $E8
>45
>46    * Zero page locations:
>47
>48    GOWARM    =    0          ;Set up by cold start
>49    GOSTROUT  =    3          ; but not used anywhere.
>50    USR       =    $A
>51    CHARAC    =    $D
>52    ENDCHR    =    $E
>53    PNTR      =    $F
>54    NUMDIM    =    $F         ;Used in array rtns
>55    DIMFLG    =    $10
>56    VALTYP    =    $11        ;$:VALTYP=$FF; %:VALTYP+1=$80
>57    DATAFLG   =    $13        ;Used in PARSE
>58    GARFLG    =    DATAFLG    ;Used in GARBAG
>59    SUBFLG    =    $14
>60    INPUTFLG  =    $15        ;Has $40 for GET, $98 for READ
>61    CPRMASK   =    $16        ;Receives CPRTYP in FRMEVL
>62    SIGNFLG   =    $16        ;Flags sign in TAN
>63    SHAPEL    =    $1A
>64    SHAPEH    =    $1B
>65    HCOLOR1   =    $1C
>66    COUNTH    =    $1D
>67    CH        =    $24
>68    GBASL     =    $26
>69    GBASH     =    $27
>70    H2        =    $2C
>71    V2        =    $2D
>72    HMASK     =    $30
>73    INVFLG    =    $32
>74    PROMPT    =    $33
>75    A1L       =    $3C
>76    A1H       =    $3D
>77    A2L       =    $3E
>78    A2H       =    $3F
>79    LINNUM    =    $50
>80    TEMPPT    =    $52
```

```
>81    LASTPT    =    $53
>82    TEMPST    =    $55
>83    INDEX     =    $5E
>84    DEST      =    $60
>85    RESULT    =    $62
>86    TXTTAB    =    $67
>87    VARTAB    =    $69
>88    ARYTAB    =    $6B
>89    STREND    =    $6D
>90    FRETOP    =    $6F
>91    FRESPC    =    $71
>92    MEMSIZ    =    $73
>93    CURLIN    =    $75
>94    OLDLIN    =    $77
>95    OLDTEXT   =    $79
>96    DATLIN    =    $7B
>97    DATPTR    =    $7D
>98    INPTR     =    $7F
>99    VARNAM    =    $81          ;$:VARNAM+1 -; %:VARNAM -
>100   VARPNT    =    $83
>101   FORPNT    =    $85
>102   TXPSV     =    $87          ;Used in INPUT
>103   LASTOP    =    $87          ;Scratch flag used in FRMEVL
>104   CPRTYP    =    $89          ;>,=,< flag in FRMEVL
>105   FNCNAM    =    $8A
>106   DSCPTR    =    $8C
>107   DSCLEN    =    $8F          ;Used in GARBAG
>108   JMPADRS   =    $90
>109   LENGTH    =    $91          ;Used in GARBAG
>110   ARYPNT    =    $94          ;Used in GARBAG
>111   HIGHDS    =    $94
>112   HIGHTR    =    $96
>113   INDX      =    $99          ;Used by array rtns
>114   LOWTR     =    $9B
>115   DSCTMP    =    $9D
>116   VPNT      =    $A0          ;Temp var ptr
>117   EXTRASV   =    $92          ;FP extra precision
>118   TEMP1     =    $93          ;Save areas for FAC
>119   TEMP2     =    $98
>120   TEMP3     =    $8A
>121   TMPEXP    =    $99          ;Used in FIN (EVAL)
>122   EXPON     =    $9A          ; "
>123   DPFLG     =    $9B          ;Flags dec pnt in FIN
>124   EXPSGN    =    $9C          ; "  sign of exp in FIN
>125   FAC       =    $9D          ;Primary floating pnt acc
>126   SERLEN    =    $A3          ;Holds length of series-1
>127   FPGEN     =    $A4
>128   ARG       =    $A5          ;Secondary fp acc
>129   FACSGN    =    FAC+5        ;Holds unpacked sign
>130   ARGSGN    =    ARG+5
>131   SGNCPR    =    $AB          ;Flags opp sign in FP rout.
>132   EXTRAFAC  =    $AC          ;FP precision
>133   SERPNT    =    $AD          ;Pntr to series data in FP
>134   STRNG1    =    $AB
>135   STRNG2    =    $AD
>136   PRGEND    =    $AF
>137   CHRGET    =    $B1
```

```
>138  CHRGOT   =    $B7
>139  TXTPTR   =    $B8
>140  RNDSEED  =    $C9
>141  DXL      =    $D0
>142  DXH      =    $D1
>143  DY       =    $D2
>144  QDRNT    =    $D3
>145  EL       =    $D4
>146  EH       =    $D5
>147  LOCK     =    $D6          ;Prevents user access if -
>148  ERRFLG   =    $D8
>149  ERRLIN   =    $DA
>150  ERRPOS   =    $DC
>151  ERRNUM   =    $DE
>152  ERRSTK   =    $DF
>153  XOL      =    $E0
>154  XOH      =    $E1
>155  YO       =    $E2
>156  HCOLORZ  =    $E4
>157  HNDX     =    $E5
>158  HPAG     =    $E6
>159  SCALEZ   =    $E7
>160  SHAPEPNT =    $E8
>161  COLCOUNT =    $EA
>162  FIRST    =    $F0
>163  SPEEDZ   =    $F1          ;Output speed
>164  TRCFLG   =    $F2
>165  ORMASK   =    $F3          ;Has $40 for flash
>166  TXTPSV   =    $F4
>167  CURLSV   =    $F6
>168  REMSTK   =    $F8
>169  ROTZ     =    $F9
>170  * $FF is also used by the string out rtns.
>171
>172  * Apple stuff:
>173
>174  STACK    =    $100
>175  IN       =    $200
>176  AMPER    =    $3F5
>177  KEY      =    $C000
>178  TXTCLR   =    $C050
>179  MIXCLR   =    $C052
>180  MIXSET   =    $C053
>181  LOWSCR   =    $C054
>182  HISCR    =    $C055
>183  LORES    =    $C056
>184  HIRES    =    $C057
>185  HOME     =    $FC58
>186  RD2BIT   =    $FCFA
>187  GETLN    =    $FD6A
>188  RDKEY    =    $FD0C
>189  WRITE    =    $FECD
>190  COUT     =    $FDED
>191  MONWAIT  =    $FCA8
>192  SCRN     =    $F871
>193  PREAD    =    $FB1E
>194  INPORT   =    $FE8B
```

```
                        >195  OUTPORT  =     $FE95
                        >196  MONPLOT  =     $F800
                        >197  HLINE    =     $F819
                        >198  VLINE    =     $F828
                        >199  SETCOL   =     $F864
                        >200  TABV     =     $FB5B
                        >201  SETGR    =     $FB40
                        >202  SETTXT   =     $FB39
                        >203  MONREAD  =     $FEFD
                        >204  MONREAD2 =     $FF02
                        >205
D000: 6F D8             >206  CMDTABL  DA    END-1
D002: 65 D7             >207           DA    FOR-1
D004: F8 DC             >208           DA    NEXT-1
D006: 94 D9             >209           DA    DATA-1
D008: B1 DB             >210           DA    INPUT-1
D00A: 30 F3             >211           DA    DEL-1
D00C: D8 DF             >212           DA    DIM-1
D00E: E1 DB             >213           DA    READ-1
D010: 8F F3             >214           DA    GR-1
D012: 98 F3             >215           DA    TEXT-1
D014: E4 F1             >216           DA    PRNU-1
D016: DD F1             >217           DA    INNU-1
D018: D4 F1             >218           DA    CALL-1
D01A: 24 F2             >219           DA    PLOT-1
D01C: 31 F2             >220           DA    HLIN-1
D01E: 40 F2             >221           DA    VLIN-1
D020: D7 F3             >222           DA    HGR2-1
D022: E1 F3             >223           DA    HGR-1
D024: E8 F6             >224           DA    HCOLOR-1
D026: FD F6             >225           DA    HPLOT-1
D028: 68 F7             >226           DA    DRAW-1
D02A: 6E F7             >227           DA    XDRAW-1
D02C: E6 F7             >228           DA    HTAB-1
D02E: 57 FC             >229           DA    HOME-1
D030: 20 F7             >230           DA    ROT-1
D032: 26 F7             >231           DA    SCALE-1
D034: 74 F7             >232           DA    SHLOAD-1
D036: 6C F2             >233           DA    TRACE-1
D038: 6E F2             >234           DA    NOTRACE-1
D03A: 72 F2             >235           DA    NORMAL-1
D03C: 76 F2             >236           DA    INVERSE-1
D03E: 7F F2             >237           DA    FLASH-1
D040: 4E F2             >238           DA    COLOR-1
D042: 6A D9             >239           DA    POP-1
D044: 55 F2             >240           DA    VTAB-1
D046: 85 F2             >241           DA    HIMEM-1
D048: A5 F2             >242           DA    LOMEM-1
D04A: CA F2             >243           DA    ONERR-1
D04C: 17 F3             >244           DA    RESUME-1
D04E: BB F3             >245           DA    RECALL-1
D050: 9E F3             >246           DA    STORE-1
D052: 61 F2             >247           DA    SPEED-1
D054: 45 DA             >248           DA    LET-1
D056: 3D D9             >249           DA    GOTO-1
D058: 11 D9             >250           DA    RUN-1
D05A: C8 D9             >251           DA    IF-1
```

```
D05C: 48 D8     >252            DA   RESTORE-1
D05E: F4 03     >253            DA   AMPER-1
D060: 20 D9     >254            DA   GOSUB-1
D062: 6A D9     >255            DA   POP-1        ;RETURN
D064: DB D9     >256            DA   REM-1
D066: 6D D8     >257            DA   STOP-1
D068: EB D9     >258            DA   ONGOTO-1
D06A: 83 E7     >259            DA   WAIT-1
D06C: C8 D8     >260            DA   LOAD-1
D06E: AF D8     >261            DA   SAVE-1
D070: 12 E3     >262            DA   DEF-1
D072: 7A E7     >263            DA   POKE-1
D074: D4 DA     >264            DA   PRINT-1
D076: 95 D8     >265            DA   CONT-1
D078: A4 D6     >266            DA   LIST-1
D07A: 69 D6     >267            DA   CLEAR-1
D07C: 9F DB     >268            DA   GET-1
D07E: 48 D6     >269            DA   NEW-1
D080: 90 EB     >270  UNFNC     DA   SGN
D082: 23 EC     >271            DA   INT
D084: AF EB     >272            DA   ABS
D086: 0A 00     >273            DA   USR
D088: DE E2     >274            DA   FRE
D08A: 12 D4     >275            DA   ERROR       ;SCRN done special
D08C: CD DF     >276            DA   PDL
D08E: FF E2     >277            DA   POS
D090: 8D EE     >278            DA   SQR
D092: AE EF     >279            DA   RND
D094: 41 E9     >280            DA   LOG
D096: 09 EF     >281            DA   EXP
D098: EA EF     >282            DA   COS
D09A: F1 EF     >283            DA   SIN
D09C: 3A F0     >284            DA   TAN
D09E: 9E F0     >285            DA   ATN
D0A0: 64 E7     >286            DA   PEEK
D0A2: D6 E6     >287            DA   LEN
D0A4: C5 E3     >288            DA   STR
D0A6: 07 E7     >289            DA   VAL
D0A8: E5 E6     >290            DA   ASC
D0AA: 46 E6     >291            DA   CHRSTR
D0AC: 5A E6     >292            DA   LEFTSTR
D0AE: 86 E6     >293            DA   RIGHTSTR
D0B0: 91 E6     >294            DA   MIDSTR
                >295
                >296  * The hex #s are for preference testing:
                >297
D0B2: 79        >298  MATHTBL   HEX  79
D0B3: C0 E7     >299            DA   FADDT-1
D0B5: 79        >300            HEX  79
D0B6: A9 E7     >301            DA   FSUBT-1
D0B8: 7B        >302            HEX  7B
D0B9: 81 E9     >303            DA   FMULTT-1
D0BB: 7B        >304            HEX  7B
D0BC: 68 EA     >305            DA   FDIVT-1
D0BE: 7D        >306            HEX  7D
D0BF: 96 EE     >307            DA   FPWRT-1
D0C1: 50        >308            HEX  50
```

```
D0C2: 54 DF     >309            DA   AND-1
D0C4: 46        >310            HEX  46
D0C5: 4E DF     >311            DA   OR-1
D0C7: 7F        >312  MINUS     HEX  7F
  C8: CF EE     >313            DA   NEGOP-1
D0CA: 7F        >314  EQUAL     HEX  7F
D0CB: 97 DE     >315            DA   EQUOP-1
D0CD: 64        >316  PLUS      HEX  64
D0CE: 64 DF     >317            DA   POSOP-1
                >318
D0D0: 45 4E C4  >319  TOKTABL   DCI  'END'
D0D3: 46 4F D2  >320            DCI  'FOR'
D0D6: 4E 45 58
D0D9: D4        >321            DCI  'NEXT'
D0DA: 44 41 54
D0DD: C1        >322            DCI  'DATA'
D0DE: 49 4E 50
D0E1: 55 D4     >323            DCI  'INPUT'
D0E3: 44 45 CC  >324            DCI  'DEL'
D0E6: 44 49 CD  >325            DCI  'DIM'
D0E9: 52 45 41
D0EC: C4        >326            DCI  'READ'
D0ED: 47 D2     >327            DCI  'GR'
D0EF: 54 45 58
D0F2: D4        >328            DCI  'TEXT'
D0F3: 50 52 A3  >329            DCI  'PR#'
D0F6: 49 4E A3  >330            DCI  'IN#'
D0F9: 43 41 4C
D0FC: CC        >331            DCI  'CALL'
  FD: 50 4C 4F
  00: D4        >332            DCI  'PLOT'
D101: 48 4C 49
D104: CE        >333            DCI  'HLIN'
D105: 56 4C 49
D108: CE        >334            DCI  'VLIN'
D109: 48 47 52
D10C: B2        >335            DCI  'HGR2'
D10D: 48 47 D2  >336            DCI  'HGR'
D110: 48 43 4F
D113: 4C 4F 52
D116: BD        >337            DCI  'HCOLOR='
D117: 48 50 4C
D11A: 4F D4     >338            DCI  'HPLOT'
D11C: 44 52 41
D11F: D7        >339            DCI  'DRAW'
D120: 58 44 52
D123: 41 D7     >340            DCI  'XDRAW'
D125: 48 54 41
D128: C2        >341            DCI  'HTAB'
D129: 48 4F 4D
D12C: C5        >342            DCI  'HOME'
D12D: 52 4F 54
D130: BD        >343            DCI  'ROT='
D131: 53 43 41
D134: 4C 45 BD  >344            DCI  'SCALE='
  37: 53 48 4C
D13A: 4F 41 C4  >345            DCI  'SHLOAD'
```

```
D13D: 54 52 41
D140: 43 C5     >346            DCI   'TRACE'
D142: 4E 4F 54
D145: 52 41 43
  48: C5        >347            DCI   'NOTRACE'
D149: 4E 4F 52
D14C: 4D 41 CC >348            DCI   'NORMAL'
D14F: 49 4E 56
D152: 45 52 53
D155: C5        >349            DCI   'INVERSE'
D156: 46 4C 41
D159: 53 C8     >350            DCI   'FLASH'
D15B: 43 4F 4C
D15E: 4F 52 BD >351            DCI   'COLOR='
D161: 50 4F D0 >352            DCI   'POP'
D164: 56 54 41
D167: C2        >353            DCI   'VTAB'
D168: 48 49 4D
D16B: 45 4D BA >354            DCI   'HIMEM:'
D16E: 4C 4F 4D
D171: 45 4D BA >355            DCI   'LOMEM:'
D174: 4F 4E 45
D177: 52 D2     >356            DCI   'ONERR'
D179: 52 45 53
D17C: 55 4D C5 >357            DCI   'RESUME'
D17F: 52 45 43
D182: 41 4C CC >358            DCI   'RECALL'
D185: 53 54 4F
D188: 52 C5     >359            DCI   'STORE'
D18A: 53 50 45
D18D: 45 44 BD >360            DCI   'SPEED='
D190: 4C 45 D4 >361            DCI   'LET'
D193: 47 4F 54
D196: CF        >362            DCI   'GOTO'
D197: 52 55 CE >363            DCI   'RUN'
D19A: 49 C6     >364            DCI   'IF'
D19C: 52 45 53
D19F: 54 4F 52
D1A2: C5        >365            DCI   'RESTORE'
D1A3: A6        >366            ASC   "&"
D1A4: 47 4F 53
D1A7: 55 C2     >367            DCI   'GOSUB'
D1A9: 52 45 54
D1AC: 55 52 CE >368            DCI   'RETURN'
D1AF: 52 45 CD >369            DCI   'REM'
D1B2: 53 54 4F
D1B5: D0        >370            DCI   'STOP'
D1B6: 4F CE     >371            DCI   'ON'
D1B8: 57 41 49
D1BB: D4        >372            DCI   'WAIT'
D1BC: 4C 4F 41
D1BF: C4        >373            DCI   'LOAD'
D1C0: 53 41 56
D1C3: C5        >374            DCI   'SAVE'
D1C4: 44 45 C6 >375            DCI   'DEF'
  C7: 50 4F 4B
D1CA: C5        >376            DCI   'POKE'
```

```
D1CB: 50 52 49
D1CE: 4E D4     >377            DCI  'PRINT'
D1D0: 43 4F 4E
D1D3: D4        >378            DCI  'CONT'
D1D4: 4C 49 53
D1D7: D4        >379            DCI  'LIST'
D1D8: 43 4C 45
D1DB: 41 D2     >380            DCI  'CLEAR'
D1DD: 47 45 D4  >381            DCI  'GET'
D1E0: 4E 45 D7  >382            DCI  'NEW'
D1E3: 54 41 42
D1E6: A8        >383            DCI  'TAB('
D1E7: 54 CF     >384            DCI  'TO'
D1E9: 46 CE     >385            DCI  'FN'
D1EB: 53 50 43
D1EE: A8        >386            DCI  'SPC('
D1EF: 54 48 45
D1F2: CE        >387            DCI  'THEN'
D1F3: 41 D4     >388            DCI  'AT'
D1F5: 4E 4F D4  >389            DCI  'NOT'
D1F8: 53 54 45
D1FB: D0        >390            DCI  'STEP'
D1FC: AB        >391            ASC  "+"
D1FD: AD        >392            ASC  "-"
D1FE: AA        >393            ASC  "*"
D1FF: AF        >394            ASC  "/"
D200: DE        >395            ASC  "^"
D201: 41 4E C4  >396            DCI  'AND'
D204: 4F D2     >397            DCI  'OR'
D206: BE        >398            ASC  ">"
D207: BD        >399            ASC  "="
D208: BC        >400            ASC  "<"
D209: 53 47 CE  >401            DCI  'SGN'
D20C: 49 4E D4  >402            DCI  'INT'
D20F: 41 42 D3  >403            DCI  'ABS'
D212: 55 53 D2  >404            DCI  'USR'
D215: 46 52 C5  >405            DCI  'FRE'
D218: 53 43 52
D21B: 4E A8     >406            DCI  'SCRN('
D21D: 50 44 CC  >407            DCI  'PDL'
D220: 50 4F D3  >408            DCI  'POS'
D223: 53 51 D2  >409            DCI  'SQR'
D226: 52 4E C4  >410            DCI  'RND'
D229: 4C 4F C7  >411            DCI  'LOG'
D22C: 45 58 D0  >412            DCI  'EXP'
D22F: 43 4F D3  >413            DCI  'COS'
D232: 53 49 CE  >414            DCI  'SIN'
D235: 54 41 CE  >415            DCI  'TAN'
D238: 41 54 CE  >416            DCI  'ATN'
D23B: 50 45 45
D23E: CB        >417            DCI  'PEEK'
D23F: 4C 45 CE  >418            DCI  'LEN'
D242: 53 54 52
D245: A4        >419            DCI  'STR$'
D246: 56 41 CC  >420            DCI  'VAL'
D249: 41 53 C3  >421            DCI  'ASC'
D24C: 43 48 52
```

```
D24F: A4          >422            DCI   'CHR$'
D250: 4C 45 46
D253: 54 A4       >423            DCI   'LEFT$'
D255: 52 49 47
  58: 48 54 A4    >424            DCI   'RIGHT$'
D25B: 4D 49 44
D25E: A4          >425            DCI   'MID$'
D25F: 00          >426            BRK
                  >427
                  >428  ERRMSG    =     *
D260: 4E 45 58
D263: 54 20 57
D266: 49 54 48
D269: 4F 55 54
D26C: 20 46 4F
D26F: D2          >429  NXwoFOR   DCI   'NEXT WITHOUT FOR'
D270: 53 59 4E
D273: 54 41 D8    >430  SYNTXERR DCI   'SYNTAX'
D276: 52 45 54
D279: 55 52 4E
D27C: 20 57 49
D27F: 54 48 4F
D282: 55 54 20
D285: 47 4F 53
D288: 55 C2       >431  RTNwoGSB DCI   'RETURN WITHOUT GOSUB'
D28A: 4F 55 54
D28D: 20 4F 46
D290: 20 44 41
D293: 54 C1       >432  OofDATA   DCI   'OUT OF DATA'
D295: 49 4C 4C
D298: 45 47 41
D29B: 4C 20 51
D29E: 55 41 4E
D2A1: 54 49 54
D2A4: D9          >433  ILLQUAN   DCI   'ILLEGAL QUANTITY'
D2A5: 4F 56 45
D2A8: 52 46 4C
D2AB: 4F D7       >434  OVFLOW    DCI   'OVERFLOW'
D2AD: 4F 55 54
D2B0: 20 4F 46
D2B3: 20 4D 45
D2B6: 4D 4F 52
D2B9: D9          >435  OofMEM    DCI   'OUT OF MEMORY'
D2BA: 55 4E 44
D2BD: 45 46 27
D2C0: 44 20 53
D2C3: 54 41 54
D2C6: 45 4D 45
D2C9: 4E D4       >436  UNDSTAT   DCI   *UNDEF'D STATEMENT*
D2CB: 42 41 44
D2CE: 20 53 55
D2D1: 42 53 43
D2D4: 52 49 50
D2D7: D4          >437  BADSUBS   DCI   'BAD SUBSCRIPT'
D2D8: 52 45 44
  DB: 49 4D 27
D2DE: 44 20 41
```

```
D2E1: 52 52 41
D2E4: D9        >438  REdimARR DCI  *REDIM'D ARRAY*
D2E5: 44 49 56
D2E8: 49 53 49
D2EB: 4F 4E 20
D2EE: 42 59 20
D2F1: 5A 45 52
D2F4: CF        >439  DIVbyZRO DCI  'DIVISION BY ZERO'
D2F5: 49 4C 4C
D2F8: 45 47 41
D2FB: 4C 20 44
D2FE: 49 52 45
D301: 43 D4     >440  ILLDIR   DCI  'ILLEGAL DIRECT'
D303: 54 59 50
D306: 45 20 4D
D309: 49 53 4D
D30C: 41 54 43
D30F: C8        >441  TYPEMISS DCI  'TYPE MISMATCH'
D310: 53 54 52
D313: 49 4E 47
D316: 20 54 4F
D319: 4F 20 4C
D31C: 4F 4E C7  >442  STRtoLNG DCI  'STRING TOO LONG'
D31F: 46 4F 52
D322: 4D 55 4C
D325: 41 20 54
D328: 4F 4F 20
D32B: 43 4F 4D
D32E: 50 4C 45
D331: D8        >443  FORMtoCX DCI  'FORMULA TOO COMPLEX'
D332: 43 41 4E
D335: 27 54 20
D338: 43 4F 4E
D33B: 54 49 4E
D33E: 55 C5     >444  CANTCON  DCI  *CAN'T CONTINUE*
D340: 55 4E 44
D343: 45 46 27
D346: 44 20 46
D349: 55 4E 43
D34C: 54 49 4F
D34F: CE        >445  UNDFUNC  DCI  *UNDEF'D FUNCTION*
                >446
D350: 20 45 52
D353: 52 4F 52  >447  ERRIN    ASC  ' ERROR'
D356: 07 00     >448           HEX  0700
D358: 20 49 4E
D35B: 20        >449  INMSG    ASC  ' IN '
D35C: 00        >450           BRK
                >451
D35D: 0D        >452  BREAKIN  HEX  0D
D35E: 42 52 45
D361: 41 4B     >453           ASC  'BREAK'
D363: 07 00     >454           HEX  0700
                >455
D365: BA        >456  GTFORPNT TSX             ;Search through stack
D366: E8        >458           INX
D367: E8        >458           INX
```

```
D368: E8        >458             INX
D369: E8        >458             INX
D36A: BD 01 01  >460  FNDFOR     LDA  STACK+1,X
D36D: C9 81     >461             CMP  #for
D36F: D0 21     >462             BNE  RET1
D371: A5 86     >463             LDA  FORPNT+1
D373: D0 0A     >464             BNE  SAMEFOR?    ;Taken if var specified
D375: BD 02 01  >465             LDA  STACK+2,X   ;Get FOR var pointer
D378: 85 85     >466             STA  FORPNT
D37A: BD 03 01  >467             LDA  STACK+3,X
D37D: 85 86     >468             STA  FORPNT+1
D37F: DD 03 01  >469  SAMEFOR?   CMP  STACK+3,X   ;Compare FOR var adrs
D382: D0 07     >470             BNE  NXFOR       ;Branch if not same
D384: A5 85     >471             LDA  FORPNT
D386: DD 02 01  >472             CMP  STACK+2,X
D389: F0 07     >473             BEQ  RET1
D38B: 8A        >474  NXFOR      TXA              ;Not correct FOR,
D38C: 18        >475             CLC              ; set up to look at next.
D38D: 69 12     >476             ADC  #$12
D38F: AA        >477             TAX
D390: D0 D8     >478             BNE  FNDFOR
D392: 60        >479  RET1       RTS
                >480
D393: 20 E3 D3  >481  BLTU       JSR  REASON      ;Is there room?
D396: 85 6D     >482             STA  STREND      ;Set top of array storage to A,Y
D398: 84 6E     >483             STY  STREND+1
                >484
                >485  * Set up to move upwards LOWTR through HIGHDS-1
                >486  * to just below HIGHDS:
                >487
D39A: 38        >488  BLTU2      SEC
                >489             >>>  SB.HIGHTR;LOWTR;INDEX
D39B: A5 96     >489             LDA  HIGHTR
D39D: E5 9B     >489             SBC  LOWTR
D39F: 85 5E     >489             STA  INDEX
                >489             <<<
D3A1: A8        >490             TAY
D3A2: A5 97     >491             LDA  HIGHTR+1
D3A4: E5 9C     >492             SBC  LOWTR+1
D3A6: AA        >493             TAX
D3A7: E8        >494             INX
D3A8: 98        >495             TYA
D3A9: F0 23     >496             BEQ  NXPAG       ;Taken if < 1 page
D3AB: A5 96     >497             LDA  HIGHTR
D3AD: 38        >498             SEC
D3AE: E5 5E     >499             SBC  INDEX
D3B0: 85 96     >500             STA  HIGHTR
D3B2: B0 03     >501             BCS  SETEND
D3B4: C6 97     >502             DEC  HIGHTR+1
D3B6: 38        >503             SEC
                >504  SETEND     >>>  SB.HIGHDS;INDEX;HIGHDS
D3B7: A5 94     >504             LDA  HIGHDS
D3B9: E5 5E     >504             SBC  INDEX
D3BB: 85 94     >504             STA  HIGHDS
                >504             <<<
D3BD: B0 08     >505             BCS  NXBYT
D3BF: C6 95     >506             DEC  HIGHDS+1
```

```
D3C1: 90 04     >507             BCC  NXBYT
D3C3: B1 96     >508  MVBYT      LDA  (HIGHTR),Y ;Now do the move
D3C5: 91 94     >509             STA  (HIGHDS),Y
D3C7: 88        >510  NXBYT      DEY
D3C8: D0 F9     >511             BNE  MVBYT
D3CA: B1 96     >512             LDA  (HIGHTR),Y
D3CC: 91 94     >513             STA  (HIGHDS),Y
D3CE: C6 97     >514  NXPAG      DEC  HIGHTR+1
D3D0: C6 95     >515             DEC  HIGHDS+1
D3D2: CA        >516             DEX             ;Another page to move?
D3D3: D0 F2     >517             BNE  NXBYT
D3D5: 60        >518             RTS
                >519
D3D6: 0A        >520  CHKMEM     ASL             ;Used by FOR, GOSUB, FRMEVL
D3D7: 69 36     >521             ADC  #$36       ; to make sure there is
D3D9: B0 35     >522             BCS  MEMERR     ; room in stack.
D3DB: 85 5E     >523             STA  INDEX
D3DD: BA        >524             TSX
D3DE: E4 5E     >525             CPX  INDEX
D3E0: 90 2E     >526             BCC  MEMERR
D3E2: 60        >527             RTS
                >528
D3E3: C4 70     >529  REASON     CPY  FRETOP+1   ;Check that A,Y < FRETOP
D3E5: 90 28     >530             BCC  RET2       ;Return if so.
D3E7: D0 04     >531             BNE  RS1        ;Clean shop if not.
D3E9: C5 6F     >532             CMP  FRETOP
D3EB: 90 22     >533             BCC  RET2
D3ED: 48        >534  RS1        PHA             ;Save A,Y and TEMP1 & TEMP2
D3EE: A2 09     >535             LDX  #FAC-TEMP1-1
D3F0: 98        >536             TYA
D3F1: 48        >537  RS2        PHA
D3F2: B5 93     >538             LDA  TEMP1,X
D3F4: CA        >539             DEX
D3F5: 10 FA     >540             BPL  RS2
D3F7: 20 84 E4  >541             JSR  GARBAG     ;Collection time
D3FA: A2 F7     >542             LDX  #TEMP1-FAC+1
D3FC: 68        >543  RS3        PLA             ;Restore TEMP1 & 2 and A,Y.
D3FD: 95 9D     >544             STA  FAC,X
D3FF: E8        >545             INX
D400: 30 FA     >546             BMI  RS3
D402: 68        >547             PLA
D403: A8        >548             TAY
D404: 68        >549             PLA             ;Is there room now?
D405: C4 70     >550             CPY  FRETOP+1
D407: 90 06     >551             BCC  RET2       ;Return if so
D409: D0 05     >552             BNE  MEMERR     ;Memory error if not.
D40B: C5 6F     >553             CMP  FRETOP
D40D: B0 01     >554             BCS  MEMERR
D40F: 60        >555  RET2       RTS
                >556
D410: A2 4D     >557  MEMERR     LDX  #OofMEM-ERRMSG
D412: 24 D8     >558  ERROR      BIT  ERRFLG
D414: 10 03     >559             BPL  DOERRMSG
D416: 4C E9 F2  >560             JMP  HANDLERR
                >561
D419: 20 FB DA  >562  DOERRMSG   JSR  CRDO
D41C: 20 5A DB  >563             JSR  OUTQUES
```

```
D41F: BD 60 D2 >564  ERLUP     LDA  ERRMSG,X
D422: 48       >565           PHA
D423: 20 5C DB >566           JSR  OUTDO
D426: E8       >567           INX
  27: 68       >568           PLA
D428: 10 F5    >569           BPL  ERLUP
D42A: 20 83 D6 >570           JSR  STKINI
D42D: A9 50    >571           LDA  #ERRIN
D42F: A0 D3    >572           LDY  #>ERRIN
D431: 20 3A DB >573  PRNTIN?   JSR  STROUT
D434: A4 76    >574           LDY  CURLIN+1
D436: C8       >575           INY
D437: F0 03    >576           BEQ  RESTART
D439: 20 19 ED >577           JSR  INPRT
               >578
D43C: 20 FB DA >579  RESTART   JSR  CRDO
D43F: A2 DD    >580           LDX  #"]"
D441: 20 2E D5 >581           JSR  INLIN2
D444: 86 B8    >582           STX  TXTPTR
D446: 84 B9    >583           STY  TXTPTR+1
D448: 46 D8    >584           LSR  ERRFLG
D44A: 20 B1 00 >585           JSR  CHRGET
D44D: AA       >586           TAX
D44E: F0 EC    >587           BEQ  RESTART
D450: A2 FF    >588           LDX  #$FF        ;Set direct mode flag
D452: 86 76    >589           STX  CURLIN+1    ; = high byte of CURLIN.
D454: 90 06    >590           BCC  NXLIN       ;Branch if line # given
D456: 20 59 D5 >591           JSR  GETIN       ;Otherwise parse
D459: 4C 05 D8 >592           JMP  TRACE?      ;and act on command.
               >593  NXLIN     >>>  TRX.PRGEND;VARTAB
 45C: A6 AF    >593           LDX  PRGEND
D45E: 86 69    >593           STX  VARTAB
D460: A6 B0    >593           LDX  PRGEND+1
D462: 86 6A    >593           STX  VARTAB+1
               >593           <<<
D464: 20 0C DA >594           JSR  LINGET      ;Get line #
D467: 20 59 D5 >595           JSR  GETIN       ;and parse input
D46A: 84 0F    >596           STY  PNTR        ;Save index to input buffer
D46C: 20 1A D6 >597           JSR  FNDLIN      ;Is line there now?
D46F: 90 44    >598           BCC  NEWLN?      ;Branch if not
D471: A0 01    >599           LDY  #1          ;If line is there, delete it.
D473: B1 9B    >600           LDA  (LOWTR),Y
D475: 85 5F    >601           STA  INDEX+1
               >602           >>>  TR.VARTAB;INDEX
D477: A5 69    >602           LDA  VARTAB
D479: 85 5E    >602           STA  INDEX
               >602           <<<
               >603           >>>  TR.LOWTR+1;DEST+1
D47B: A5 9C    >603           LDA  LOWTR+1
D47D: 85 61    >603           STA  DEST+1
               >603           <<<
D47F: A5 9B    >604           LDA  LOWTR
D481: 88       >605           DEY
D482: F1 9B    >606           SBC  (LOWTR),Y
D484: 18       >607           CLC
 485: 65 69    >608           ADC  VARTAB
D487: 85 69    >609           STA  VARTAB
```

```
D489: 85 60     >610            STA  DEST
D48B: A5 6A     >611            LDA  VARTAB+1
D48D: 69 FF     >612            ADC  #$FF
D48F: 85 6A     >613            STA  VARTAB+1
D491: E5 9C     >614            SBC  LOWTR+1
D493: AA        >615            TAX
D494: 38        >616            SEC
D495: A5 9B     >617            LDA  LOWTR
D497: E5 69     >618            SBC  VARTAB
D499: A8        >619            TAY
D49A: B0 03     >620            BCS  NL1
D49C: E8        >621            INX
D49D: C6 61     >622            DEC  DEST+1
D49F: 18        >623  NL1       CLC
D4A0: 65 5E     >624            ADC  INDEX
D4A2: 90 03     >625            BCC  MVDWN
D4A4: C6 5F     >626            DEC  INDEX+1
D4A6: 18        >627            CLC
D4A7: B1 5E     >628  MVDWN     LDA  (INDEX),Y   ;Move rest of program
D4A9: 91 60     >629            STA  (DEST),Y    ;to deleted line's place.
D4AB: C8        >630            INY
D4AC: D0 F9     >631            BNE  MVDWN
D4AE: E6 5F     >632            INC  INDEX+1
D4B0: E6 61     >633            INC  DEST+1
D4B2: CA        >634            DEX
D4B3: D0 F2     >635            BNE  MVDWN
D4B5: AD 00 02  >636  NEWLN?    LDA  IN          ;Line # alone?
D4B8: F0 38     >637            BEQ  LINKSET     ;Skip to LINKSET if so.
                >638            >>>  TRAY.MEMSIZ;FRETOP
D4BA: A5 73     >638            LDA  MEMSIZ
D4BC: A4 74     >638            LDY  MEMSIZ+1
D4BE: 85 6F     >638            STA  FRETOP
D4C0: 84 70     >638            STY  FRETOP+1
                >638            <<<
D4C2: A5 69     >639            LDA  VARTAB      ;Set up memory move to
D4C4: 85 96     >640            STA  HIGHTR      ;insert new line.
D4C6: 65 0F     >641            ADC  PNTR
D4C8: 85 94     >642            STA  HIGHDS
D4CA: A4 6A     >643            LDY  VARTAB+1
D4CC: 84 97     >644            STY  HIGHTR+1
D4CE: 90 01     >645            BCC  MVPRG
D4D0: C8        >646            INY
D4D1: 84 95     >647  MVPRG     STY  HIGHDS+1
D4D3: 20 93 D3  >648            JSR  BLTU        ;Do the move
                >649            >>>  TRAY.LINNUM;IN-2
D4D6: A5 50     >649            LDA  LINNUM
D4D8: A4 51     >649            LDY  LINNUM+1
D4DA: 8D FE 01  >649            STA  IN-2
D4DD: 8C FF 01  >649            STY  IN-2+1
                >649            <<<
                >650            >>>  TRAY.STREND;VARTAB
D4E0: A5 6D     >650            LDA  STREND
D4E2: A4 6E     >650            LDY  STREND+1
D4E4: 85 69     >650            STA  VARTAB
D4E6: 84 6A     >650            STY  VARTAB+1
                >650            <<<
D4E8: A4 0F     >651            LDY  PNTR
```

```
D4EA: B9 FB 01 >652  INSRTLIN LDA  IN-5,Y      ;Insert new line
D4ED: 88      >653            DEY
D4EE: 91 9B   >654            STA  (LOWTR),Y
D4F0: D0 F8   >655            BNE  INSRTLIN
D4F2: 20 65 D6 >656  LINKSET  JSR  SETPTRS
              >657            >>>  TRAY.TXTTAB;INDEX
D4F5: A5 67   >657            LDA  TXTTAB
D4F7: A4 68   >657            LDY  TXTTAB+1
D4F9: 85 5E   >657            STA  INDEX
D4FB: 84 5F   >657            STY  INDEX+1
              >657            <<<
D4FD: 18      >658            CLC
D4FE: A0 01   >659  NXLINK    LDY  #1
D500: B1 5E   >660            LDA  (INDEX),Y
D502: D0 0B   >661            BNE  PUTLINK
              >662            >>>  TRDB.VARTAB;PRGEND
              >662            >>>  TR.VARTAB;PRGEND
D504: A5 69   >662            LDA  VARTAB
D506: 85 AF   >662            STA  PRGEND
              >662            <<<
              >662            >>>  TR.VARTAB+1;PRGEND+1
D508: A5 6A   >662            LDA  VARTAB+1
D50A: 85 B0   >662            STA  PRGEND+1
              >662            <<<
              >662            <<<
D50C: 4C 3C D4 >663           JMP  RESTART
D50F: A0 04   >664  PUTLINK   LDY  #4          ;Set up links
D511: C8      >665  FINDEOL   INY
D512: B1 5E   >666            LDA  (INDEX),Y
D514: D0 FB   >667            BNE  FINDEOL
D516: C8      >668            INY
D517: 98      >669            TYA
D518: 65 5E   >670            ADC  INDEX
D51A: AA      >671            TAX
D51B: A0 00   >672            LDY  #0
D51D: 91 5E   >673            STA  (INDEX),Y
D51F: A5 5F   >674            LDA  INDEX+1
D521: 69 00   >675            ADC  #0
D523: C8      >676            INY
D524: 91 5E   >677            STA  (INDEX),Y
D526: 86 5E   >678            STX  INDEX
D528: 85 5F   >679            STA  INDEX+1
D52A: 90 D2   >680            BCC  NXLINK
              >681
D52C: A2 80   >682  INLIN     LDX  #$80
D52E: 86 33   >683  INLIN2    STX  PROMPT
D530: 20 6A FD >684           JSR  GETLN
D533: E0 EF   >685            CPX  #$EF
D535: 90 02   >686            BCC  GDBUFS
D537: A2 EF   >687            LDX  #$EF        ;Terminate line at $EF chrs
D539: A9 00   >688  GDBUFS    LDA  #0          ;Set up eol marker
D53B: 9D 00 02 >689           STA  IN,X
D53E: 8A      >690            TXA
D53F: F0 0B   >691            BEQ  NOI
D541: BD FF 01 >692  STRIP    LDA  IN-1,X      ;Convert to + ascii
D544: 29 7F   >693            AND  #$7F
D546: 9D FF 01 >694           STA  IN-1,X
```

```
D549: CA        >695             DEX
D54A: D0 F5     >696             BNE  STRIP
D54C: A9 00     >697  NOI        LDA  #0
D54E: A2 FF     >698             LDX  #IN-1
  50: A0 01     >699             LDY  #>IN-1
D552: 60        >700             RTS
                >701
D553: 20 0C FD  >702  INCHR      JSR  RDKEY
D556: 29 7F     >703             AND  #$7F
D558: 60        >704             RTS
                >705
D559: A6 B8     >706  GETIN      LDX  TXTPTR
D55B: CA        >707             DEX
D55C: A0 04     >708             LDY  #4
D55E: 84 13     >709             STY  DATAFLG
D560: 24 D6     >710             BIT  LOCK        ;Program protected?
D562: 10 08     >711             BPL  PARSE
D564: 68        >712             PLA              ;If so, ignore input
D565: 68        >713             PLA              ;and run program again.
D566: 20 65 D6  >714             JSR  SETPTRS
D569: 4C D2 D7  >715             JMP  NEWSTT
                >716
D56C: E8        >717  PARSE      INX
D56D: BD 00 02  >718  NXCHR      LDA  IN,X
D570: 24 13     >719             BIT  DATAFLG
D572: 70 04     >720             BVS  SE          ;Branch if DATA stmnt
D574: C9 20     >721             CMP  #' '
D576: F0 F4     >722             BEQ  PARSE
D578: 85 0E     >723  SE         STA  ENDCHR
D57A: C9 22     >724             CMP  #'"'
D57C: F0 74     >725             BEQ  SHIN
D57E: 70 4D     >726             BVS  PUTIN       ;Branch if DATA stmnt
D580: C9 3F     >727             CMP  #'?'
D582: D0 04     >728             BNE  TOK?
D584: A9 BA     >729             LDA  #print
D586: D0 45     >730             BNE  PUTIN       ;Always
D588: C9 30     >731  TOK?       CMP  #'0'
D58A: 90 04     >732             BLT  ISTOK?
D58C: C9 3C     >733             CMP  #'<'
D58E: 90 3D     >734             BLT  PUTIN
D590: 84 AD     >735  ISTOK?     STY  STRNG2
D592: A9 D0     >736             LDA  #TOKTABL-$100
D594: 85 9D     >737             STA  FAC
D596: A9 CF     >738             LDA  #>TOKTABL-$100
D598: 85 9E     >739             STA  FAC+1
D59A: A0 00     >740             LDY  #0
D59C: 84 0F     >741             STY  PNTR        ;Holds current token-$80
D59E: 88        >742             DEY
D59F: 86 B8     >743             STX  TXTPTR
D5A1: CA        >744             DEX
D5A2: C8        >745  NY         INY
D5A3: D0 02     >746             BNE  NX
D5A5: E6 9E     >747             INC  FAC+1
D5A7: E8        >748  NX         INX
D5A8: BD 00 02  >749  LIN        LDA  IN,X
D5AB: C9 20     >750             CMP  #' '        ;Skip spaces
D5AD: F0 F8     >751             BEQ  NX
```

```
D5AF: 38        >752            SEC
D5B0: F1 9D     >753            SBC  (FAC),Y     ;Does it match keyword?
D5B2: F0 EE     >754            BEQ  NY          ;Next chr if so
D5B4: C9 80     >755            CMP  #$80        ;Match last keyword chr?
  B6: D0 41     >756            BNE  SKIPTOK     ;Skip to next token if not
D5B8: 05 0F     >757            ORA  PNTR        ;Get token
D5BA: C9 C5     >758            CMP  #at
D5BC: D0 0D     >759            BNE  PUTTOK
D5BE: BD 01 02  >760            LDA  IN+1,X
D5C1: C9 4E     >761            CMP  #'N'        ;Preferance to ATN
D5C3: F0 34     >762            BEQ  SKIPTOK
D5C5: C9 4F     >763            CMP  #'O'        ;Preferance to TO
D5C7: F0 30     >764            BEQ  SKIPTOK
D5C9: A9 C5     >765            LDA  #at
D5CB: A4 AD     >766  PUTTOK    LDY  STRNG2
D5CD: E8        >767  PUTIN     INX
D5CE: C8        >768            INY
D5CF: 99 FB 01  >769            STA  IN-5,Y
D5D2: B9 FB 01  >770            LDA  IN-5,Y
D5D5: F0 39     >771            BEQ  DONE
D5D7: 38        >772            SEC
D5D8: E9 3A     >773            SBC  #':'
D5DA: F0 04     >774            BEQ  SSF         ;Reset DATAFLG at stmnt end
D5DC: C9 49     >775            CMP  #data-':'
D5DE: D0 02     >776            BNE  REM?
D5E0: 85 13     >777  SSF       STA  DATAFLG
D5E2: 38        >778  REM?      SEC
D5E3: E9 78     >779            SBC  #rem-':'
D5E5: D0 86     >780            BNE  NXCHR
D5E7: 85 0E     >781            STA  ENDCHR      ;Clear literal flag
D5E9: BD 00 02  >782  SHFTIN    LDA  IN,X
D5EC: F0 DF     >783            BEQ  PUTIN
D5EE: C5 0E     >784            CMP  ENDCHR
D5F0: F0 DB     >785            BEQ  PUTIN
D5F2: C8        >786  SHIN      INY
D5F3: 99 FB 01  >787            STA  IN-5,Y
D5F6: E8        >788            INX
D5F7: D0 F0     >789            BNE  SHFTIN      ;Loop till literal done
D5F9: A6 B8     >790  SKIPTOK   LDX  TXTPTR
D5FB: E6 0F     >791            INC  PNTR        ;Next token
D5FD: B1 9D     >792  SK2       LDA  (FAC),Y     ;Skip over current keyword
D5FF: C8        >793            INY
D600: D0 02     >794            BNE  PLU?
D602: E6 9E     >795            INC  FAC+1
D604: 0A        >796  PLU?      ASL
D605: 90 F6     >797            BCC  SK2         ;Loop till keyword skipped
D607: B1 9D     >798            LDA  (FAC),Y
D609: D0 9D     >799            BNE  LIN         ;Loop till keyword table done
D60B: BD 00 02  >800            LDA  IN,X        ;Not keyword
D60E: 10 BB     >801            BPL  PUTTOK      ;Always
D610: 99 FD 01  >802  DONE      STA  IN-3,Y      ;Point TXTPTR at IN-1
D613: C6 B9     >803            DEC  TXTPTR+1
D615: A9 FF     >804            LDA  #$FF
D617: 85 B8     >805            STA  TXTPTR
D619: 60        >806            RTS
                >807
                >808  * Search program for line whose # is now in LINNUM.
```

```
                        >809  * On exit: carry is set if found, clear if not,
                        >810  * LOWTR points to line if found, to next one if not.
                        >811
D61A: A5 67             >812  FNDLIN   LDA  TXTTAB
  1C: A6 68             >813          LDX  TXTTAB+1
D61E: A0 01             >814  FL1      LDY  #1
D620: 85 9B             >815          STA  LOWTR
D622: 86 9C             >816          STX  LOWTR+1
D624: B1 9B             >817          LDA  (LOWTR),Y
D626: F0 1F             >818          BEQ  NOSUCH      ;Branch if end of program
D628: C8                >819          INY
D629: C8                >820          INY
D62A: A5 51             >821          LDA  LINNUM+1
D62C: D1 9B             >822          CMP  (LOWTR),Y
D62E: 90 18             >823          BCC  RET3        ;If not found
D630: F0 03             >824          BEQ  FL2
D632: 88                >825          DEY
D633: D0 09             >826          BNE  GETLINK
D635: A5 50             >827  FL2      LDA  LINNUM
D637: 88                >828          DEY
D638: D1 9B             >829          CMP  (LOWTR),Y
D63A: 90 0C             >830          BCC  RET3        ;Past line, not found
D63C: F0 0A             >831          BEQ  RET3        ;If found
D63E: 88                >832  GETLINK  DEY
D63F: B1 9B             >833          LDA  (LOWTR),Y
D641: AA                >834          TAX
D642: 88                >835          DEY
D643: B1 9B             >836          LDA  (LOWTR),Y
D645: B0 D7             >837          BCS  FL1         ;Always
D647: 18                >838  NOSUCH   CLC
D648: 60                >839  RET3     RTS
                        >840
D649: D0 FD             >841  NEW      BNE  RET3
D64B: A9 00             >842  SCRTCH   LDA  #0
D64D: 85 D6             >843          STA  LOCK
D64F: A8                >844          TAY
D650: 91 67             >845          STA  (TXTTAB),Y
D652: C8                >846          INY
D653: 91 67             >847          STA  (TXTTAB),Y
D655: A5 67             >848          LDA  TXTTAB
D657: 69 02             >849          ADC  #2          ;Carry is indeterminate
D659: 85 69             >850          STA  VARTAB
D65B: 85 AF             >851          STA  PRGEND
D65D: A5 68             >852          LDA  TXTTAB+1
D65F: 69 00             >853          ADC  #0
D661: 85 6A             >854          STA  VARTAB+1
D663: 85 B0             >855          STA  PRGEND+1
D665: 20 97 D6          >856  SETPTRS  JSR  STXTPT
D668: A9 00             >857          LDA  #0
D66A: D0 2A             >858  CLEAR    BNE  RET4
                        >859  CLEARC   >>>  TRAY.MEMSIZ;FRETOP
D66C: A5 73             >859          LDA  MEMSIZ
D66E: A4 74             >859          LDY  MEMSIZ+1
D670: 85 6F             >859          STA  FRETOP
D672: 84 70             >859          STY  FRETOP+1
                        >859          <<<
                        >860          >>>  TRAY.VARTAB;ARYTAB
```

```
D674: A5 69     >860            LDA  VARTAB
D676: A4 6A     >860            LDY  VARTAB+1
D678: 85 6B     >860            STA  ARYTAB
D67A: 84 6C     >860            STY  ARYTAB+1
                >860            <<<
D67C: 85 6D     >861            STA  STREND
D67E: 84 6E     >862            STY  STREND+1
D680: 20 49 D8  >863            JSR  RESTORE
D683: A2 55     >864  STKINI    LDX  #TEMPST
D685: 86 52     >865            STX  TEMPPT
D687: 68        >866            PLA
D688: A8        >867            TAY
D689: 68        >868            PLA
D68A: A2 F8     >869            LDX  #$F8
D68C: 9A        >870            TXS
D68D: 48        >871            PHA
D68E: 98        >872            TYA
D68F: 48        >873            PHA
D690: A9 00     >874            LDA  #0
D692: 85 7A     >875            STA  OLDTEXT+1
D694: 85 14     >876            STA  SUBFLG
D696: 60        >877  RET4      RTS
                >878
D697: 18        >879  STXTPT    CLC
                >880            >>>  AD.TXTTAB;#$FF;TXTPTR
D698: A5 67     >880            LDA  TXTTAB
D69A: 69 FF     >880            ADC  #$FF
D69C: 85 B8     >880            STA  TXTPTR
                >880            <<<
                >881            >>>  AD.TXTTAB+1;#$FF;TXTPTR+1
D69E: A5 68     >881            LDA  TXTTAB+1
D6A0: 69 FF     >881            ADC  #$FF
D6A2: 85 B9     >881            STA  TXTPTR+1
                >881            <<<
D6A4: 60        >882            RTS
                >883
D6A5: 90 0A     >884  LIST      BCC  STRTRNG     ;Line # specified?
D6A7: F0 08     >885            BEQ  STRTRNG     ;No
D6A9: C9 C9     >886            CMP  #minus      ;Start range at 0 if so
D6AB: F0 04     >887            BEQ  STRTRNG
D6AD: C9 2C     >888            CMP  #','
D6AF: D0 E5     >889            BNE  RET4
D6B1: 20 0C DA  >890  STRTRNG   JSR  LINGET      ;Set LINNUM to start of rng
D6B4: 20 1A D6  >891            JSR  FNDLIN      ;Point LOWTR to 1st line
D6B7: 20 B7 00  >892            JSR  CHRGOT      ;Range specified?
D6BA: F0 10     >893            BEQ  MAINLST     ;Branch if not
D6BC: C9 C9     >894            CMP  #minus
D6BE: F0 04     >895            BEQ  ENDRNG
D6C0: C9 2C     >896            CMP  #','
D6C2: D0 84     >897            BNE  RET3
D6C4: 20 B1 00  >898  ENDRNG    JSR  CHRGET      ;Update TXTPTR
D6C7: 20 0C DA  >899            JSR  LINGET      ;Set LINNUM to end rng
D6CA: D0 CA     >900            BNE  RET4        ;Branch if syntax err
D6CC: 68        >901  MAINLST   PLA              ;Pop rtn adrs
D6CD: 68        >902            PLA
  CE: A5 50     >903            LDA  LINNUM
D6D0: 05 51     >904            ORA  LINNUM+1
```

```
D6D2: D0 06     >905            BNE  NXLST
D6D4: A9 FF     >906            LDA  #$FF        ;Max end range
D6D6: 85 50     >907            STA  LINNUM
D6D8: 85 51     >908            STA  LINNUM+1
D6DA: A0 01     >909  NXLST     LDY  #1
D6DC: B1 9B     >910            LDA  (LOWTR),Y   ;High byte of link
D6DE: F0 44     >911            BEQ  LISTED
D6E0: 20 58 D8  >912            JSR  ISCNTC      ;Check for control C
D6E3: 20 FB DA  >913            JSR  CRDO
D6E6: C8        >914            INY
D6E7: B1 9B     >915            LDA  (LOWTR),Y   ;Get line number
D6E9: AA        >916            TAX
D6EA: C8        >917            INY
D6EB: B1 9B     >918            LDA  (LOWTR),Y
D6ED: C5 51     >919            CMP  LINNUM+1
D6EF: D0 04     >920            BNE  LSTD?
D6F1: E4 50     >921            CPX  LINNUM
D6F3: F0 02     >922            BEQ  LST1LIN
D6F5: B0 2D     >923  LSTD?     BCS  LISTED
D6F7: 84 85     >924  LST1LIN   STY  FORPNT
D6F9: 20 24 ED  >925            JSR  LINPRT      ;Print X,A
D6FC: A9 20     >926            LDA  #' '
D6FE: A4 85     >927  LISTLOOP  LDY  FORPNT
D700: 29 7F     >928            AND  #$7F
D702: 20 5C DB  >929  SENDCHR   JSR  OUTDO
D705: A5 24     >930            LDA  CH
D707: C9 21     >931            CMP  #33         ;If over 33, do CR
D709: 90 07     >932            BCC  NCR
D70B: 20 FB DA  >933            JSR  CRDO
D70E: A9 05     >934            LDA  #5          ; and tab over 5
D710: 85 24     >935            STA  CH
D712: C8        >936  NCR       INY
D713: B1 9B     >937            LDA  (LOWTR),Y
D715: D0 1D     >938            BNE  TOKEN?
D717: A8        >939            TAY              ;At end of line, get link
D718: B1 9B     >940            LDA  (LOWTR),Y
D71A: AA        >941            TAX
D71B: C8        >942            INY
D71C: B1 9B     >943            LDA  (LOWTR),Y
D71E: 86 9B     >944            STX  LOWTR       ;Point to next line
D720: 85 9C     >945            STA  LOWTR+1
D722: D0 B6     >946            BNE  NXLST
D724: A9 0D     >947  LISTED    LDA  #$D         ;CR and out
D726: 20 5C DB  >948            JSR  OUTDO
D729: 4C D2 D7  >949            JMP  NEWSTT
                >950
D72C: C8        >951  GETCHR    INY              ;Pick up chr from table
D72D: D0 02     >952            BNE  GC
D72F: E6 9E     >953            INC  FAC+1
D731: B1 9D     >954  GC        LDA  (FAC),Y
D733: 60        >955            RTS
                >956
D734: 10 CC     >957  TOKEN?    BPL  SENDCHR     ;Branch if not token
D736: 38        >958            SEC
D737: E9 7F     >959            SBC  #$7F        ;Make index to table
D739: AA        >960            TAX
D73A: 84 85     >961            STY  FORPNT      ;Save line pointer
```

```
D73C: A0 D0     >962            LDY  #TOKTABL-$100
D73E: 84 9D     >963            STY  FAC          ;Point FAC to table
D740: A0 CF     >964            LDY  #>TOKTABL-$100
D742: 84 9E     >965            STY  FAC+1
  44: A0 FF     >966            LDY  #$FF
D746: CA        >967 SKPTK      DEX               ;Count tokens versa X
D747: F0 07     >968            BEQ  PRTOK
D749: 20 2C D7  >969 TOKLP      JSR  GETCHR
D74C: 10 FB     >970            BPL  TOKLP
D74E: 30 F6     >971            BMI  SKPTK
D750: A9 20     >972 PRTOK      LDA  #' '         ;Token found, send space
D752: 20 5C DB  >973            JSR  OUTDO
D755: 20 2C D7  >974 TOKLUP     JSR  GETCHR       ; then token
D758: 30 05     >975            BMI  TOKDONE
D75A: 20 5C DB  >976            JSR  OUTDO
D75D: D0 F6     >977            BNE  TOKLUP
D75F: 20 5C DB  >978 TOKDONE    JSR  OUTDO        ;Send last chr of token
D762: A9 20     >979            LDA  #' '         ;Send end space
D764: D0 98     >980            BNE  LISTLOOP     ;Back to actual line
                >981
                >982  * FOR places following 18 bytes on stack:
                >983  *  TXTPTR
                >984  *  Line number
                >985  *  FOR variable value (5 byte FP #)
                >986  *  STEP sign
                >987  *  STEP value (5 byte)
                >988  *  FORPNT (pointer to var1)
                >989  *  FOR token
                >990
766: A9 80      >991 FOR        LDA  #$80
768: 85 14      >992            STA  SUBFLG       ;Subscripts not allowed
D76A: 20 46 DA  >993            JSR  LET
D76D: 20 65 D3  >994            JSR  GTFORPNT     ;Is this FOR var1 active?
D770: D0 05     >995            BNE  FOR2         ;Branch if not
D772: 8A        >996            TXA               ;If so, cancel it and
D773: 69 0F     >997            ADC  #$F          ; all subsequent ones.
D775: AA        >998            TAX
D776: 9A        >999            TXS
D777: 68        >1000 FOR2      PLA
D778: 68        >1001           PLA
D779: A9 09     >1002           LDA  #9
D77B: 20 D6 D3  >1003           JSR  CHKMEM
D77E: 20 A3 D9  >1004           JSR  DATAN        ;Point to next statement
D781: 18        >1005           CLC               ; and push this address.
D782: 98        >1006           TYA
D783: 65 B8     >1007           ADC  TXTPTR
D785: 48        >1008           PHA
D786: A5 B9     >1009           LDA  TXTPTR+1
D788: 69 00     >1010           ADC  #0
D78A: 48        >1011           PHA
                >1012           >>>  PUSH.CURLIN
D78B: A5 76     >1012           LDA  CURLIN+1
D78D: 48        >1012           PHA
D78E: A5 75     >1012           LDA  CURLIN
D790: 48        >1012           PHA
                >1012           <<<
D791: A9 C1     >1013           LDA  #to
```

```
D793: 20 C0 DE >1014           JSR  SYNCHR
D796: 20 6A DD >1015           JSR  CHKNUM
D799: 20 67 DD >1016           JSR  FRMNUM
D79C: A5 A2    >1017           LDA  FACSGN
D79E: 09 7F    >1018           ORA  #$7F
D7A0: 25 9E    >1019           AND  FAC+1
D7A2: 85 9E    >1020           STA  FAC+1
D7A4: A9 AF    >1021           LDA  #STEP       ;Set up for return
D7A6: A0 D7    >1022           LDY  #>STEP      ; to STEP
D7A8: 85 5E    >1023           STA  INDEX
D7AA: 84 5F    >1024           STY  INDEX+1
D7AC: 4C 20 DE >1025           JMP  PUSHFAC     ;Returns to STEP
D7AF: A9 13    >1026 STEP      LDA  #ONE        ;STEP default=1
D7B1: A0 E9    >1027           LDY  #>ONE
D7B3: 20 F9 EA >1028           JSR  MOVFM
D7B6: 20 B7 00 >1029           JSR  CHRGOT
D7B9: C9 C7    >1030           CMP  #step
D7BB: D0 06    >1031           BNE  ONESTEP
D7BD: 20 B1 00 >1032           JSR  CHRGET      ;Step specified, get it
D7C0: 20 67 DD >1033           JSR  FRMNUM
D7C3: 20 82 EB >1034 ONESTEP   JSR  SIGN
D7C6: 20 15 DE >1035           JSR  PSHFACX
               >1036           >>>  PUSH.FORPNT
D7C9: A5 86    >1036           LDA  FORPNT+1
D7CB: 48       >1036           PHA
D7CC: A5 85    >1036           LDA  FORPNT
D7CE: 48       >1036           PHA
               >1036           <<<
D7CF: A9 81    >1037           LDA  #for
D7D1: 48       >1038           PHA
D7D2: BA       >1039 NEWSTT    TSX              ;Execute new statement
D7D3: 86 F8    >1040           STX  REMSTK
D7D5: 20 58 D8 >1041           JSR  ISCNTC
D7D8: A5 B8    >1042           LDA  TXTPTR
D7DA: A4 B9    >1043           LDY  TXTPTR+1
D7DC: A6 76    >1044           LDX  CURLIN+1    ;Direct mode
D7DE: E8       >1045           INX
D7DF: F0 04    >1046           BEQ  DIRCT       ;Branch if so
D7E1: 85 79    >1047           STA  OLDTEXT     ;Save TXTPTR if in program
D7E3: 84 7A    >1048           STY  OLDTEXT+1
D7E5: A0 00    >1049 DIRCT     LDY  #0
D7E7: B1 B8    >1050           LDA  (TXTPTR),Y  ;At eol?
D7E9: D0 57    >1051           BNE  COLON?      ;If not, is it a colon?
D7EB: A0 02    >1052           LDY  #2          ;If so, is link 0?
D7ED: B1 B8    >1053           LDA  (TXTPTR),Y
D7EF: 18       >1054           CLC
D7F0: F0 34    >1055           BEQ  GOEND       ;Done if link 0
D7F2: C8       >1056           INY
D7F3: B1 B8    >1057           LDA  (TXTPTR),Y
D7F5: 85 75    >1058           STA  CURLIN      ;If not done, save line #
D7F7: C8       >1059           INY
D7F8: B1 B8    >1060           LDA  (TXTPTR),Y
D7FA: 85 76    >1061           STA  CURLIN+1
D7FC: 98       >1062           TYA
               >1063           >>>  BUMP.TXTPTR ;And set up txtptr
D7FD: 65 B8    >1063           ADC  TXTPTR
D7FF: 85 B8    >1063           STA  TXTPTR
```

```
D801: 90 02     >1063           BCC  BPD
D803: E6 B9     >1063           INC  TXTPTR+1
                >1063 BPD       <<<
D805: 24 F2     >1064 TRACE?    BIT  TRCFLG      ;Trace requested?
  07: 10 14     >1065           BPL  TR1         ;Branch if not
D809: A6 76     >1066           LDX  CURLIN+1
D80B: E8        >1067           INX
D80C: F0 0F     >1068           BEQ  TR1         ;Skip if direct command
D80E: A9 23     >1069           LDA  #'#'        ;Print "#"
D810: 20 5C DB  >1070           JSR  OUTDO
D813: A6 75     >1071           LDX  CURLIN
D815: A5 76     >1072           LDA  CURLIN+1
D817: 20 24 ED  >1073           JSR  LINPRT      ;and the number
D81A: 20 57 DB  >1074           JSR  OUTSP
D81D: 20 B1 00  >1075 TR1       JSR  CHRGET      ;Get first chr of statement
D820: 20 28 D8  >1076           JSR  GOCMD       ;and start processing
D823: 4C D2 D7  >1077           JMP  NEWSTT      ;Back for more
                >1078
D826: F0 62     >1079 GOEND     BEQ  END4
D828: F0 2D     >1080 GOCMD     BEQ  RET5
D82A: E9 80     >1081 GOCMD2    SBC  #$80        ;A token?
D82C: 90 11     >1082           BCC  NOTOK       ;Branch if not
D82E: C9 40     >1083           CMP  #$40        ;"Routine" type token?
D830: B0 14     >1084           BCS  JSY         ;Syntax error if not
D832: 0A        >1085           ASL              ;If a routine token,
D833: A8        >1086           TAY              ;then place routine address
D834: B9 01 D0  >1087           LDA  CMDTABL+1,Y
D837: 48        >1088           PHA              ;on stack,
D838: B9 00 D0  >1089           LDA  CMDTABL,Y
  3B: 48        >1090           PHA
  3C: 4C B1 00  >1091           JMP  CHRGET      ;Get next chr & RTS to routine.
D83F: 4C 46 DA  >1092 NOTOK     JMP  LET         ;Must be a variable assignment
D842: C9 3A     >1093 COLON?    CMP  #':'
D844: F0 BF     >1094           BEQ  TRACE?
D846: 4C C9 DE  >1095 JSY       JMP  SYNERR
                >1096
D849: 38        >1097 RESTORE   SEC
D84A: A5 67     >1098           LDA  TXTTAB
D84C: E9 01     >1099           SBC  #1
D84E: A4 68     >1100           LDY  TXTTAB+1
D850: B0 01     >1101           BCS  SETDA
D852: 88        >1102           DEY
D853: 85 7D     >1103 SETDA     STA  DATPTR
D855: 84 7E     >1104           STY  DATPTR+1
D857: 60        >1105 RET5      RTS
                >1106
D858: AD 00 C0  >1107 ISCNTC    LDA  KEY
D85B: C9 83     >1108           CMP  #$83
D85D: F0 01     >1109           BEQ  GK
D85F: 60        >1110           RTS
D860: 20 53 D5  >1111 GK        JSR  INCHR
D863: A2 FF     >1112 ERFLG?    LDX  #$FF        ;Control C attempted
D865: 24 D8     >1113           BIT  ERRFLG
D867: 10 03     >1114           BPL  CTRC?
D869: 4C E9 F2  >1115           JMP  HANDLERR
  6C: C9 03     >1116 CTRC?     CMP  #3
                >1117
```

```
D86E: B0 01     >1118 STOP      BCS  END2
                >1119
D870: 18        >1120 END       CLC
D871: D0 3C     >1121 END2      BNE  RET6
D873: A5 B8     >1122           LDA  TXTPTR
D875: A4 B9     >1123           LDY  TXTPTR+1
D877: A6 76     >1124           LDX  CURLIN+1
D879: E8        >1125           INX
D87A: F0 0C     >1126           BEQ  END3
D87C: 85 79     >1127           STA  OLDTEXT
D87E: 84 7A     >1128           STY  OLDTEXT+1
                >1129           >>>  TRAY.CURLIN;OLDLIN
D880: A5 75     >1129           LDA  CURLIN
D882: A4 76     >1129           LDY  CURLIN+1
D884: 85 77     >1129           STA  OLDLIN
D886: 84 78     >1129           STY  OLDLIN+1
                >1129           <<<
D888: 68        >1130 END3      PLA
D889: 68        >1131           PLA
D88A: A9 5D     >1132 END4      LDA  #BREAKIN
D88C: A0 D3     >1133           LDY  #>BREAKIN
D88E: 90 03     >1134           BCC  GOSTART
D890: 4C 31 D4  >1135           JMP  PRNTIN?
D893: 4C 3C D4  >1136 GOSTART   JMP  RESTART
                >1137
D896: D0 17     >1138 CONT      BNE  RET6
D898: A2 D2     >1139           LDX  #CANTCON-ERRMSG
D89A: A4 7A     >1140           LDY  OLDTEXT+1
D89C: D0 03     >1141           BNE  CON
D89E: 4C 12 D4  >1142           JMP  ERROR
D8A1: A5 79     >1143 CON       LDA  OLDTEXT
D8A3: 85 B8     >1144           STA  TXTPTR
D8A5: 84 B9     >1145           STY  TXTPTR+1
                >1146           >>>  TRAY.OLDLIN;CURLIN
D8A7: A5 77     >1146           LDA  OLDLIN
D8A9: A4 78     >1146           LDY  OLDLIN+1
D8AB: 85 75     >1146           STA  CURLIN
D8AD: 84 76     >1146           STY  CURLIN+1
                >1146           <<<
D8AF: 60        >1147 RET6      RTS
                >1148
                >1149 SAVE      >>>  SUB.PRGEND;TXTTAB;LINNUM
D8B0: 38        >1149           SEC
                >1149           >>>  SB.PRGEND;TXTTAB;LINNUM
D8B1: A5 AF     >1149           LDA  PRGEND
D8B3: E5 67     >1149           SBC  TXTTAB
D8B5: 85 50     >1149           STA  LINNUM
                >1149           <<<
                >1149           >>>  SB.PRGEND+1;TXTTAB+1;LINNUM+1
D8B7: A5 B0     >1149           LDA  PRGEND+1
D8B9: E5 68     >1149           SBC  TXTTAB+1
D8BB: 85 51     >1149           STA  LINNUM+1
                >1149           <<<
                >1149           <<<
D8BD: 20 F0 D8  >1150           JSR  VARTIO
D8C0: 20 CD FE  >1151           JSR  WRITE
D8C3: 20 01 D9  >1152           JSR  PROGIO
```

```
D8C6: 4C CD FE >1153           JMP  WRITE
D8C9: 20 F0 D8 >1154 LOAD      JSR  VARTIO
D8CC: 20 FD FE >1155           JSR  MONREAD
               >1156           >>>  ADD.TXTTAB;LINNUM;VARTAB
 CF: 18        >1156           CLC
               >1156           >>>  AD.TXTTAB;LINNUM;VARTAB
D8D0: A5 67    >1156           LDA  TXTTAB
D8D2: 65 50    >1156           ADC  LINNUM
D8D4: 85 69    >1156           STA  VARTAB
               >1156           <<<
               >1156           >>>  AD.TXTTAB+1;LINNUM+1;VARTAB+1
D8D6: A5 68    >1156           LDA  TXTTAB+1
D8D8: 65 51    >1156           ADC  LINNUM+1
D8DA: 85 6A    >1156           STA  VARTAB+1
               >1156           <<<
               >1156           <<<
               >1157           >>>  TR.TEMPPT;LOCK
D8DC: A5 52    >1157           LDA  TEMPPT
D8DE: 85 D6    >1157           STA  LOCK
               >1157           <<<
D8E0: 20 01 D9 >1158           JSR  PROGIO
D8E3: 20 FD FE >1159           JSR  MONREAD
D8E6: 24 D6    >1160           BIT  LOCK
D8E8: 10 03    >1161           BPL  JLNK
D8EA: 4C 65 D6 >1162           JMP  SETPTRS
D8ED: 4C F2 D4 >1163 JLNK      JMP  LINKSET
               >1164
D8F0: A9 50    >1165 VARTIO    LDA  #LINNUM
D8F2: A0 00    >1166           LDY  #0
 8F4: 85 3C    >1167           STA  A1L
 8F6: 84 3D    >1168           STY  A1H
D8F8: A9 52    >1169           LDA  #TEMPPT
D8FA: 85 3E    >1170           STA  A2L
D8FC: 84 3F    >1171           STY  A2H
D8FE: 84 D6    >1172           STY  LOCK
D900: 60       >1173           RTS
               >1174
               >1175 PROGIO    >>>  TRAY.TXTTAB;A1L
D901: A5 67    >1175           LDA  TXTTAB
D903: A4 68    >1175           LDY  TXTTAB+1
D905: 85 3C    >1175           STA  A1L
D907: 84 3D    >1175           STY  A1L+1
               >1175           <<<
               >1176           >>>  TRAY.VARTAB;A2L
D909: A5 69    >1176           LDA  VARTAB
D90B: A4 6A    >1176           LDY  VARTAB+1
D90D: 85 3E    >1176           STA  A2L
D90F: 84 3F    >1176           STY  A2L+1
               >1176           <<<
D911: 60       >1177           RTS
               >1178
D912: 08       >1179 RUN       PHP
D913: C6 76    >1180           DEC  CURLIN+1
D915: 28       >1181           PLP
D916: D0 03    >1182           BNE  RUNLINE    ;Branch if line given
 18: 4C 65 D6 >1183            JMP  SETPTRS    ;"Specify" program start
D91B: 20 6C D6 >1184 RUNLINE   JSR  CLEARC     ;Clear varls
```

```
D91E: 4C 35 D9 >1185            JMP  GOLINE      ;Go to line specified
              >1186
              >1187 * GOSUB leaves following on stack:
              >1188 *  Return address (NEWSTT)
              >1189 *  TXTPTR
              >1190 *  line number
              >1191 *  GOSUB token
              >1192
D921: A9 03   >1193 GOSUB     LDA  #3
D923: 20 D6 D3 >1194          JSR  CHKMEM
              >1195           >>>  PUSH.TXTPTR
D926: A5 B9   >1195           LDA  TXTPTR+1
D928: 48      >1195           PHA
D929: A5 B8   >1195           LDA  TXTPTR
D92B: 48      >1195           PHA
              >1195           <<<
              >1196           >>>  PUSH.CURLIN
D92C: A5 76   >1196           LDA  CURLIN+1
D92E: 48      >1196           PHA
D92F: A5 75   >1196           LDA  CURLIN
D931: 48      >1196           PHA
              >1196           <<<
D932: A9 B0   >1197           LDA  #gosub
D934: 48      >1198           PHA
D935: 20 B7 00 >1199 GOLINE   JSR  CHRGOT
D938: 20 3E D9 >1200          JSR  GOTO
D93B: 4C D2 D7 >1201          JMP  NEWSTT
              >1202
D93E: 20 0C DA >1203 GOTO     JSR  LINGET      ;Get GOTO line
D941: 20 A6 D9 >1204          JSR  REMN        ;Point Y to eol
D944: A5 76   >1205           LDA  CURLIN+1    ;Is current page < GOTO page?
D946: C5 51   >1206           CMP  LINNUM+1
D948: B0 0B   >1207           BCS  GO1         ;Search from prog start if not
D94A: 98      >1208           TYA              ;Otherwise search from next line
D94B: 38      >1209           SEC
D94C: 65 B8   >1210           ADC  TXTPTR
D94E: A6 B9   >1211           LDX  TXTPTR+1
D950: 90 07   >1212           BCC  GO2
D952: E8      >1213           INX
D953: B0 04   >1214           BCS  GO2
D955: A5 67   >1215 GO1       LDA  TXTTAB      ;Get program beginning
D957: A6 68   >1216           LDX  TXTTAB+1
D959: 20 1E D6 >1217 GO2      JSR  FL1         ;Search for GOTO line
D95C: 90 1E   >1218           BCC  UNDERR      ;Error if not there
              >1219                            ;Point TXTPTR to GOTO line
              >1220           >>>  SB.LOWTR;#1;TXTPTR
D95E: A5 9B   >1220           LDA  LOWTR
D960: E9 01   >1220           SBC  #1
D962: 85 B8   >1220           STA  TXTPTR
              >1220           <<<
              >1221           >>>  SB.LOWTR+1;#0;TXTPTR+1
D964: A5 9C   >1221           LDA  LOWTR+1
D966: E9 00   >1221           SBC  #0
D968: 85 B9   >1221           STA  TXTPTR+1
              >1221           <<<
D96A: 60      >1222 RET7      RTS              ;Return to NEWSTT or GOSUB
              >1223
```

```
D96B: D0 FD     >1224 POP       BNE  RET7
D96D: A9 FF     >1225           LDA  #$FF
D96F: 85 85     >1226           STA  FORPNT      ;Bug: should be FORPNT+1
D971: 20 65 D3  >1227           JSR  GTFORPNT    ;To cancel FOR/NEXT in sub
   74: 9A       >1228           TXS
D975: C9 B0     >1229           CMP  #gosub      ;Last GOSUB found?
D977: F0 0B     >1230           BEQ  RETURN
D979: A2 16     >1231           LDX  #RTNwoGSB-ERRMSG
D97B: 2C        >1232           HEX  2C          ;Fake
D97C: A2 5A     >1233 UNDERR    LDX  #UNDSTAT-ERRMSG
D97E: 4C 12 D4  >1234           JMP  ERROR
                >1235
D981: 4C C9 DE  >1236 GSYNER    JMP  SYNERR
                >1237
D984: 68        >1238 RETURN    PLA
D985: 68        >1239           PLA
D986: C0 42     >1240           CPY  #pop*2
D988: F0 3B     >1241           BEQ  PULL3       ;Branch if a POP
D98A: 85 75     >1242           STA  CURLIN      ;Retrieve line #
D98C: 68        >1243           PLA
D98D: 85 76     >1244           STA  CURLIN+1
                >1245           >>>  PULL.TXTPTR ;and TXTPTR
D98F: 68        >1245           PLA
D990: 85 B8     >1245           STA  TXTPTR
D992: 68        >1245           PLA
D993: 85 B9     >1245           STA  TXTPTR+1
                >1245           <<<
D995: 20 A3 D9  >1246 DATA      JSR  DATAN       ;Move to next statement
D998: 98        >1247 ADDON     TYA
   99: 18       >1248           CLC
                >1249           >>>  BUMP.TXTPTR
D99A: 65 B8     >1249           ADC  TXTPTR
D99C: 85 B8     >1249           STA  TXTPTR
D99E: 90 02     >1249           BCC  BPD
D9A0: E6 B9     >1249           INC  TXTPTR+1
                >1249 BPD       <<<
D9A2: 60        >1250 RET8      RTS
                >1251
D9A3: A2 3A     >1252 DATAN     LDX  #':'        ;Get offset in Y to eol or ":"
D9A5: 2C        >1253           HEX  2C          ;Fake
D9A6: A2 00     >1254 REMN      LDX  #0          ; " to eol only.
D9A8: 86 0D     >1255           STX  CHARAC
D9AA: A0 00     >1256           LDY  #0
D9AC: 84 0E     >1257           STY  ENDCHR
D9AE: A5 0E     >1258 RM1       LDA  ENDCHR      ;Trick to count quote parity
D9B0: A6 0D     >1259           LDX  CHARAC
D9B2: 85 0D     >1260           STA  CHARAC
D9B4: 86 0E     >1261           STX  ENDCHR
D9B6: B1 B8     >1262 RM2       LDA  (TXTPTR),Y
D9B8: F0 E8     >1263           BEQ  RET8
D9BA: C5 0E     >1264           CMP  ENDCHR
D9BC: F0 E4     >1265           BEQ  RET8
D9BE: C8        >1266           INY
D9BF: C9 22     >1267           CMP  #'"'
D9C1: D0 F3     >1268           BNE  RM2
   C3: F0 E9    >1269           BEQ  RM1
                >1270
```

```
D9C5: 68        >1271 PULL3    PLA
D9C6: 68        >1272          PLA
D9C7: 68        >1273          PLA
D9C8: 60        >1274          RTS
                >1275
D9C9: 20 7B DD  >1276 IF       JSR  FRMEVL
D9CC: 20 B7 00  >1277          JSR  CHRGOT
D9CF: C9 AB     >1278          CMP  #goto
D9D1: F0 05     >1279          BEQ  TRUE?
D9D3: A9 C4     >1280          LDA  #then
D9D5: 20 C0 DE  >1281          JSR  SYNCHR
D9D8: A5 9D     >1282 TRUE?    LDA  FAC        ;Condition true or false?
D9DA: D0 05     >1283          BNE  IFTRUE     ;Branch if true
D9DC: 20 A6 D9  >1284 REM      JSR  REMN       ;Skip rest of line
D9DF: F0 B7     >1285          BEQ  ADDON      ;Always taken
                >1286
D9E1: 20 B7 00  >1287 IFTRUE   JSR  CHRGOT     ;Command or number?
D9E4: B0 03     >1288          BCS  JGOCMD     ;Branch if command
D9E6: 4C 3E D9  >1289          JMP  GOTO       ;Go if #
D9E9: 4C 28 D8  >1290 JGOCMD   JMP  GOCMD
                >1291
D9EC: 20 F8 E6  >1292 ONGOTO   JSR  GETBYT     ;Get specified # in FAC+4
D9EF: 48        >1293          PHA
D9F0: C9 B0     >1294          CMP  #gosub
D9F2: F0 04     >1295          BEQ  ONCNT
D9F4: C9 AB     >1296 GOTO?    CMP  #goto
D9F6: D0 89     >1297          BNE  GSYNER
D9F8: C6 A1     >1298 ONCNT    DEC  FAC+4      ;Counted to right one yet?
D9FA: D0 04     >1299          BNE  NXNUM      ;No, keep looking
D9FC: 68        >1300          PLA             ;Yes, retrieve cmd
D9FD: 4C 2A D8  >1301          JMP  GOCMD2     ;and go.
DA00: 20 B1 00  >1302 NXNUM    JSR  CHRGET
DA03: 20 0C DA  >1303          JSR  LINGET
DA06: C9 2C     >1304          CMP  #','
DA08: F0 EE     >1305          BEQ  ONCNT
DA0A: 68        >1306          PLA             ;Not found, so ignore
DA0B: 60        >1307 RET9     RTS
                >1308
DA0C: A2 00     >1309 LINGET   LDX  #0         ;ASC # to HEX address
DA0E: 86 50     >1310          STX  LINNUM     ;in LINNUM.
DA10: 86 51     >1311          STX  LINNUM+1
DA12: B0 F7     >1312 ASCHEX   BCS  RET9
DA14: E9 2F     >1313          SBC  #'0'-1
DA16: 85 0D     >1314          STA  CHARAC
DA18: A5 51     >1315          LDA  LINNUM+1
DA1A: 85 5E     >1316          STA  INDEX
DA1C: C9 19     >1317          CMP  #$FA/10    ;Line # too large?
DA1E: B0 D4     >1318          BCS  GOTO?      ;Get error if so.
                >1319                          ;(Note that GOTO xxxxxy
                >1320                          ; where xxxxx is between
                >1321                          ; 43776 and 44031 causes
                >1322                          ; a jump to $22D9.  GOSUBs etc
                >1323                          ; jump to other locations.)
DA20: A5 50     >1324          LDA  LINNUM
DA22: 0A        >1326          ASL
DA23: 26 5E     >1327          ROL  INDEX
DA25: 0A        >1326          ASL
```

```
DA26: 26 5E     >1327           ROL  INDEX
DA28: 65 50     >1329           ADC  LINNUM
DA2A: 85 50     >1330           STA  LINNUM
                >1331           >>>  AD.INDEX;LINNUM+1;LINNUM+1
 2C: A5 5E      >1331           LDA  INDEX
DA2E: 65 51     >1331           ADC  LINNUM+1
DA30: 85 51     >1331           STA  LINNUM+1
                >1331           <<<
DA32: 06 50     >1332           ASL  LINNUM
DA34: 26 51     >1333           ROL  LINNUM+1
                >1334           >>>  AD.LINNUM;CHARAC;LINNUM
DA36: A5 50     >1334           LDA  LINNUM
DA38: 65 0D     >1334           ADC  CHARAC
DA3A: 85 50     >1334           STA  LINNUM
                >1334           <<<
DA3C: 90 02     >1335           BCC  NXDIG
DA3E: E6 51     >1336           INC  LINNUM+1
DA40: 20 B1 00  >1337 NXDIG     JSR  CHRGET
DA43: 4C 12 DA  >1338           JMP  ASCHEX
                >1339
DA46: 20 E3 DF  >1340 LET       JSR  PTRGET
DA49: 85 85     >1341           STA  FORPNT
DA4B: 84 86     >1342           STY  FORPNT+1
DA4D: A9 D0     >1343           LDA  #equal
DA4F: 20 C0 DE  >1344           JSR  SYNCHR
                >1345           >>>  PUSH.VALTYP
DA52: A5 12     >1345           LDA  VALTYP+1
DA54: 48        >1345           PHA
DA55: A5 11     >1345           LDA  VALTYP
 57: 48         >1345           PHA
                >1345           <<<
DA58: 20 7B DD  >1346           JSR  FRMEVL
DA5B: 68        >1347           PLA
DA5C: 2A        >1348           ROL
DA5D: 20 6D DD  >1349           JSR  CHKVAL
DA60: D0 18     >1350           BNE  LETSTR      ;If a string
DA62: 68        >1351           PLA
DA63: 10 12     >1352 LET2      BPL  LETREAL
DA65: 20 72 EB  >1353           JSR  RNDB        ;Integer var
DA68: 20 0C E1  >1354           JSR  AYINT
DA6B: A0 00     >1355           LDY  #0
DA6D: A5 A0     >1356           LDA  VPNT
DA6F: 91 85     >1357           STA  (FORPNT),Y
DA71: C8        >1358           INY
DA72: A5 A1     >1359           LDA  VPNT+1
DA74: 91 85     >1360           STA  (FORPNT),Y
DA76: 60        >1361           RTS
DA77: 4C 27 EB  >1362 LETREAL   JMP  SETFOR
DA7A: 68        >1363 LETSTR    PLA
DA7B: A0 02     >1364 PUTSTR    LDY  #2
DA7D: B1 A0     >1365           LDA  (VPNT),Y
DA7F: C5 70     >1366           CMP  FRETOP+1
DA81: 90 17     >1367           BCC  COPSTR      ;Branch if not in str space
DA83: D0 07     >1368           BNE  DESC?
DA85: 88        >1369           DEY
 86: B1 A0      >1370           LDA  (VPNT),Y
DA88: C5 6F     >1371           CMP  FRETOP
```

```
DA8A: 90 0E     >1372           BCC  COPSTR
DA8C: A4 A1     >1373 DESC?     LDY  VPNT+1      ;Descriptor exist?
DA8E: C4 6A     >1374           CPY  VARTAB+1
DA90: 90 08     >1375           BCC  COPSTR
  92: D0 0D     >1376           BNE  NEWDESC
DA94: A5 A0     >1377           LDA  VPNT
DA96: C5 69     >1378           CMP  VARTAB
DA98: B0 07     >1379           BCS  NEWDESC
DA9A: A5 A0     >1380 COPSTR    LDA  VPNT
DA9C: A4 A1     >1381           LDY  VPNT+1
DA9E: 4C B7 DA  >1382           JMP  COPY
DAA1: A0 00     >1383 NEWDESC   LDY  #0
DAA3: B1 A0     >1384           LDA  (VPNT),Y
DAA5: 20 D5 E3  >1385           JSR  STRINI
                >1386           >>>  TRAY.DSCPTR;STRNG1
DAA8: A5 8C     >1386           LDA  DSCPTR
DAAA: A4 8D     >1386           LDY  DSCPTR+1
DAAC: 85 AB     >1386           STA  STRNG1
DAAE: 84 AC     >1386           STY  STRNG1+1
                >1386           <<<
DAB0: 20 D4 E5  >1387           JSR  MOVINS
DAB3: A9 9D     >1388           LDA  #FAC
DAB5: A0 00     >1389           LDY  #0
DAB7: 85 8C     >1390 COPY      STA  DSCPTR
DAB9: 84 8D     >1391           STY  DSCPTR+1
DABB: 20 35 E6  >1392           JSR  FRETMS
DABE: A0 00     >1393           LDY  #0
DAC0: B1 8C     >1395           LDA  (DSCPTR),Y
DAC2: 91 85     >1396           STA  (FORPNT),Y
  C4: C8        >1397           INY
  C5: B1 8C     >1395           LDA  (DSCPTR),Y
DAC7: 91 85     >1396           STA  (FORPNT),Y
DAC9: C8        >1397           INY
DACA: B1 8C     >1399           LDA  (DSCPTR),Y
DACC: 91 85     >1400           STA  (FORPNT),Y
DACE: 60        >1401           RTS
                >1402
DACF: 20 3D DB  >1403 PRSTRING  JSR  STRPRT
DAD2: 20 B7 00  >1404           JSR  CHRGOT
DAD5: F0 24     >1405 PRINT     BEQ  CRDO
DAD7: F0 29     >1406 PRINT2    BEQ  RET1O
DAD9: C9 C0     >1407           CMP  #tab
DADB: F0 39     >1408           BEQ  TABWHERE
DADD: C9 C3     >1409           CMP  #spc
DADF: 18        >1410           CLC
DAE0: F0 34     >1411           BEQ  TABWHERE
DAE2: C9 2C     >1412           CMP  #','
DAE4: 18        >1413           CLC              ;No purpose to this
DAE5: F0 1C     >1414           BEQ  TAB
DAE7: C9 3B     >1415           CMP  #';'
DAE9: F0 44     >1416           BEQ  NEXTCHR
DAEB: 20 7B DD  >1417           JSR  FRMEVL
DAEE: 24 11     >1418           BIT  VALTYP
DAF0: 30 DD     >1419           BMI  PRSTRING
DAF2: 20 34 ED  >1420           JSR  FOUT
  F5: 20 E7 E3  >1421           JSR  STRLIT
DAF8: 4C CF DA  >1422           JMP  PRSTRING
```

```
                    >1423
DAFB: A9 0D         >1424 CRDO      LDA  #$D
DAFD: 20 5C DB      >1425           JSR  OUTDO
DB00: 49 FF         >1426 NEGATE    EOR  #$FF
DB02: 60            >1427 RET10     RTS
                    >1428
DB03: A5 24         >1429 TAB       LDA  CH
DB05: C9 18         >1430           CMP  #$18        ;This should be $20 (bug)
DB07: 90 05         >1431           BCC  NXCLM
DB09: 20 FB DA      >1432           JSR  CRDO
DB0C: D0 21         >1433           BNE  NEXTCHR
DB0E: 69 10         >1434 NXCLM     ADC  #$10
DB10: 29 F0         >1435           AND  #$F0        ;Tabs 16, 32
DB12: 85 24         >1436           STA  CH
DB14: 90 19         >1437           BCC  NEXTCHR
DB16: 08            >1438 TABWHERE  PHP
DB17: 20 F5 E6      >1439           JSR  GTBYTC
DB1A: C9 29         >1440           CMP  #')'
DB1C: F0 03         >1441           BEQ  SPC?
DB1E: 4C C9 DE      >1442           JMP  SYNERR
DB21: 28            >1443 SPC?      PLP
DB22: 90 07         >1444           BCC  TABIT       ;Branch if SPC
DB24: CA            >1445           DEX
DB25: 8A            >1446           TXA
DB26: E5 24         >1447           SBC  CH
DB28: 90 05         >1448           BCC  NEXTCHR
DB2A: AA            >1449           TAX
DB2B: E8            >1450 TABIT     INX
DB2C: CA            >1451 NXSPC     DEX
DB2D: D0 06         >1452           BNE  DOSPC
DB2F: 20 B1 00      >1453 NEXTCHR   JSR  CHRGET
DB32: 4C D7 DA      >1454           JMP  PRINT2
DB35: 20 57 DB      >1455 DOSPC     JSR  OUTSP
DB38: D0 F2         >1456           BNE  NXSPC
DB3A: 20 E7 E3      >1457 STROUT    JSR  STRLIT
DB3D: 20 00 E6      >1458 STRPRT    JSR  FREFAC
DB40: AA            >1459           TAX
DB41: A0 00         >1460           LDY  #0
DB43: E8            >1461           INX
DB44: CA            >1462 NXCHAR    DEX
DB45: F0 BB         >1463           BEQ  RET10
DB47: B1 5E         >1464           LDA  (INDEX),Y
DB49: 20 5C DB      >1465           JSR  OUTDO
DB4C: C8            >1466           INY
DB4D: C9 0D         >1467           CMP  #$D
DB4F: D0 F3         >1468           BNE  NXCHAR
DB51: 20 00 DB      >1469           JSR  NEGATE
DB54: 4C 44 DB      >1470           JMP  NXCHAR
                    >1471
                    >1472 * Note: POKE 243,32 ($20 in $F3) will convert
                    >1473 * output to lower case.  This can be cancelled
                    >1474 * by NORMAL, INVERSE, or FLASH or POKE 243,0.
                    >1475
DB57: A9 20         >1476 OUTSP     LDA  #' '
DB59: 2C            >1477           HEX  2C          ;Fake
DB5A: A9 3F         >1478 OUTQUES   LDA  #'?'
DB5C: 09 80         >1479 OUTDO     ORA  #$80
```

```
DB5E: C9 A0     >1480           CMP  #" "        ;Control chr?
DB60: 90 02     >1481           BLT  SEND        ;Skip if so
DB62: 05 F3     >1482           ORA  ORMASK      ;Convert to flash or no change
DB64: 20 ED FD  >1483 SEND      JSR  COUT
   67: 29 7F    >1484           AND  #$7F
DB69: 48        >1485           PHA
DB6A: A5 F1     >1486           LDA  SPEEDZ
DB6C: 20 A8 FC  >1487           JSR  MONWAIT
DB6F: 68        >1488           PLA
DB70: 60        >1489           RTS
                >1490
DB71: A5 15     >1491 INPUTERR  LDA  INPUTFLG
DB73: F0 12     >1492           BEQ  RESPERR     ;Taken if INPUT
DB75: 30 04     >1493           BMI  READERR     ;Taken if READ
DB77: A0 FF     >1494           LDY  #$FF        ;From a GET
DB79: D0 04     >1495           BNE  ERLIN
DB7B: A5 7B     >1496 READERR   LDA  DATLIN
DB7D: A4 7C     >1497           LDY  DATLIN+1
DB7F: 85 75     >1498 ERLIN     STA  CURLIN
DB81: 84 76     >1499           STY  CURLIN+1
DB83: 4C C9 DE  >1500           JMP  SYNERR
DB86: 68        >1501 INPERR    PLA
DB87: 24 D8     >1502 RESPERR   BIT  ERRFLG
DB89: 10 05     >1503           BPL  DOREENT
DB8B: A2 FE     >1504           LDX  #$FE        ;Bad responce
DB8D: 4C E9 F2  >1505           JMP  HANDLERR
DB90: A9 EF     >1506 DOREENT   LDA  #REENT
DB92: A0 DC     >1507           LDY  #>REENT
DB94: 20 3A DB  >1508           JSR  STROUT
                >1509           >>>  TRAY.OLDTEXT;TXTPTR
  97: A5 79     >1509           LDA  OLDTEXT
DB99: A4 7A     >1509           LDY  OLDTEXT+1
DB9B: 85 B8     >1509           STA  TXTPTR
DB9D: 84 B9     >1509           STY  TXTPTR+1
                >1509           <<<
DB9F: 60        >1510           RTS
                >1511
DBA0: 20 06 E3  >1512 GET       JSR  ERRDIR
DBA3: A2 01     >1513           LDX  #IN+1       ;Simulate input
DBA5: A0 02     >1514           LDY  #>IN+1
DBA7: A9 00     >1515           LDA  #0
DBA9: 8D 01 02  >1516           STA  IN+1
DBAC: A9 40     >1517           LDA  #$40        ;Set up INPUTFLG
DBAE: 20 EB DB  >1518           JSR  MAININP
DBB1: 60        >1519           RTS
                >1520
DBB2: C9 22     >1521 INPUT     CMP  #'"'        ;Check for optional
DBB4: D0 0E     >1522           BNE  QOUT        ;input string.
DBB6: 20 81 DE  >1523           JSR  STRTXT
DBB9: A9 3B     >1524           LDA  #';'
DBBB: 20 C0 DE  >1525           JSR  SYNCHR
DBBE: 20 3D DB  >1526           JSR  STRPRT
DBC1: 4C C7 DB  >1527           JMP  DIR?
DBC4: 20 5A DB  >1528 QOUT      JSR  OUTQUES     ;No string, print "?"
DBC7: 20 06 E3  >1529 DIR?      JSR  ERRDIR
  CA: A9 2C     >1530           LDA  #','
DBCC: 8D FF 01  >1531           STA  IN-1
```

```
DBCF: 20 2C D5 >1532           JSR  INLIN
DBD2: AD 00 02 >1533           LDA  IN
DBD5: C9 03    >1534           CMP  #3          ;Control C?
DBD7: D0 10    >1535           BNE  ZF
  D9: 4C 63 D8 >1536           JMP  ERFLG?
               >1537
DBDC: 20 5A DB >1538 NXIN      JSR  OUTQUES
DBDF: 4C 2C D5 >1539           JMP  INLIN
DBE2: A6 7D    >1540 READ      LDX  DATPTR
DBE4: A4 7E    >1541           LDY  DATPTR+1
DBE6: A9 98    >1542           LDA  #$98
DBE8: 2C       >1543           HEX  2C          ;Trick to branch to MAININP
DBE9: A9 00    >1544 ZF        LDA  #0
DBEB: 85 15    >1545 MAININP   STA  INPUTFLG
DBED: 86 7F    >1546           STX  INPTR
DBEF: 84 80    >1547           STY  INPTR+1
DBF1: 20 E3 DF >1548 NXINP     JSR  PTRGET
DBF4: 85 85    >1549           STA  FORPNT
DBF6: 84 86    >1550           STY  FORPNT+1
               >1551           >>>  TRAY.TXTPTR;TXPSV
DBF8: A5 B8    >1551           LDA  TXTPTR
DBFA: A4 B9    >1551           LDY  TXTPTR+1
DBFC: 85 87    >1551           STA  TXPSV
DBFE: 84 88    >1551           STY  TXPSV+1
               >1551           <<<
               >1552           >>>  TRXY.INPTR;TXTPTR
DC00: A6 7F    >1552           LDX  INPTR
DC02: A4 80    >1552           LDY  INPTR+1
DC04: 86 B8    >1552           STX  TXTPTR
 C06: 84 B9    >1552           STY  TXTPTR+1
               >1552           <<<
DC08: 20 B7 00 >1553           JSR  CHRGOT
DC0B: D0 1E    >1554           BNE  INSTART
DC0D: 24 15    >1555           BIT  INPUTFLG
DC0F: 50 0E    >1556           BVC  SNDQ?       ;Branch if not GET
DC11: 20 0C FD >1557           JSR  RDKEY       ;GET it
DC14: 29 7F    >1558           AND  #$7F
DC16: 8D 00 02 >1559           STA  IN
DC19: A2 FF    >1560           LDX  #IN-1
DC1B: A0 01    >1561           LDY  #>IN-1
DC1D: D0 08    >1562           BNE  STXP
               >1563
DC1F: 30 7F    >1564 SNDQ?     BMI  FINDATA
DC21: 20 5A DB >1565           JSR  OUTQUES
DC24: 20 DC DB >1566           JSR  NXIN
DC27: 86 B8    >1567 STXP      STX  TXTPTR
DC29: 84 B9    >1568           STY  TXTPTR+1
DC2B: 20 B1 00 >1569 INSTART   JSR  CHRGET
DC2E: 24 11    >1570           BIT  VALTYP
DC30: 10 31    >1571           BPL  NUMIN
DC32: 24 15    >1572           BIT  INPUTFLG
DC34: 50 09    >1573           BVC  PUTCHR      ;Branch if not GET
DC36: E8       >1574           INX
DC37: 86 B8    >1575           STX  TXTPTR
DC39: A9 00    >1576           LDA  #0
  3B: 85 0D    >1577           STA  CHARAC
DC3D: F0 0C    >1578           BEQ  PENCHR
```

```
DC3F: 85 0D     >1579 PUTCHR   STA  CHARAC
DC41: C9 22     >1580          CMP  #'"'
DC43: F0 07     >1581          BEQ  PECHR
DC45: A9 3A     >1582          LDA  #':'
DC47: 85 0D     >1583          STA  CHARAC
DC49: A9 2C     >1584          LDA  #','
DC4B: 18        >1585 PENCHR   CLC
DC4C: 85 0E     >1586 PECHR    STA  ENDCHR
DC4E: A5 B8     >1587          LDA  TXTPTR
DC50: A4 B9     >1588          LDY  TXTPTR+1
DC52: 69 00     >1589          ADC  #0          ;Skip quote, if there
DC54: 90 01     >1590          BCC  SKP
DC56: C8        >1591          INY
DC57: 20 ED E3  >1592 SKP      JSR  STRLT2
DC5A: 20 3D E7  >1593          JSR  POINT
DC5D: 20 7B DA  >1594          JSR  PUTSTR
DC60: 4C 72 DC  >1595          JMP  WNX
DC63: 48        >1596 NUMIN    PHA
DC64: AD 00 02  >1597          LDA  IN          ;From DATA?
DC67: F0 30     >1598          BEQ  INPFIN      ;Branch if so
DC69: 68        >1599 DATIN    PLA
DC6A: 20 4A EC  >1600          JSR  FIN         ;Get FP number at TXTPNT
DC6D: A5 12     >1601          LDA  VALTYP+1
DC6F: 20 63 DA  >1602          JSR  LET2        ;Put in var1
DC72: 20 B7 00  >1603 WNX      JSR  CHRGOT
DC75: F0 07     >1604          BEQ  SWPNT       ;Branch if input done?
DC77: C9 2C     >1605          CMP  #','        ;Comma in input?
DC79: F0 03     >1606          BEQ  SWPNT
DC7B: 4C 71 DB  >1607          JMP  INPUTERR    ;Nothing else will do
                >1608 SWPNT    >>>  TRAY.TXTPTR;INPTR
DC7E: A5 B8     >1608          LDA  TXTPTR
DC80: A4 B9     >1608          LDY  TXTPTR+1
DC82: 85 7F     >1608          STA  INPTR
DC84: 84 80     >1608          STY  INPTR+1
                >1608          <<<
                >1609          >>>  TRAY.TXPSV;TXTPTR
DC86: A5 87     >1609          LDA  TXPSV
DC88: A4 88     >1609          LDY  TXPSV+1
DC8A: 85 B8     >1609          STA  TXTPTR
DC8C: 84 B9     >1609          STY  TXTPTR+1
                >1609          <<<
DC8E: 20 B7 00  >1610          JSR  CHRGOT
DC91: F0 33     >1611          BEQ  INPDONE     ;If statement not done
DC93: 20 BE DE  >1612          JSR  CHKCOM      ; program must have comma.
DC96: 4C F1 DB  >1613          JMP  NXINP       ;Get next input
DC99: A5 15     >1614 INPFIN   LDA  INPUTFLG
DC9B: D0 CC     >1615          BNE  DATIN
DC9D: 4C 86 DB  >1616          JMP  INPERR
                >1617
DCA0: 20 A3 D9  >1618 FINDATA  JSR  DATAN       ;Get offset to next statement
DCA3: C8        >1619          INY
DCA4: AA        >1620          TAX              ;End of line?
DCA5: D0 12     >1621          BNE  NXS         ;Branch if ":"
DCA7: A2 2A     >1622          LDX  #0ofDATA-ERRMSG
DCA9: C8        >1623          INY
DCAA: B1 B8     >1624          LDA  (TXTPTR),Y ;End of program?
DCAC: F0 5F     >1625          BEQ  GERR        ;Error if so
```

```
DCAE: C8        >1626           INY
DCAF: B1 B8     >1627           LDA  (TXTPTR),Y ;Get next line #
DCB1: 85 7B     >1628           STA  DATLIN
DCB3: C8        >1629           INY
  B4: B1 B8     >1630           LDA  (TXTPTR),Y
DCB6: C8        >1631           INY
DCB7: 85 7C     >1632           STA  DATLIN+1
DCB9: B1 B8     >1633 NXS       LDA  (TXTPTR),Y ;Get 1st token of statement
DCBB: AA        >1634           TAX
DCBC: 20 98 D9  >1635           JSR  ADDON      ;Update TXTPTR
DCBF: E0 83     >1636           CPX  #data
DCC1: D0 DD     >1637           BNE  FINDATA    ;Loop till DATA found
DCC3: 4C 2B DC  >1638           JMP  INSTART    ;Found DATA token
DCC6: A5 7F     >1639 INPDONE   LDA  INPTR      ;No more input requested
DCC8: A4 80     >1640           LDY  INPTR+1
DCCA: A6 15     >1641           LDX  INPUTFLG
DCCC: 10 03     >1642           BPL  NTD
DCCE: 4C 53 D8  >1643           JMP  SETDA      ;If from DATA
DCD1: A0 00     >1644 NTD       LDY  #0
DCD3: B1 7F     >1645           LDA  (INPTR),Y  ;Extra input?
DCD5: F0 07     >1646           BEQ  RET11
DCD7: A9 DF     >1647           LDA  #EXIG      ;Error if so
DCD9: A0 DC     >1648           LDY  #>EXIG
DCDB: 4C 3A DB  >1649           JMP  STROUT
DCDE: 60        >1650 RET11     RTS
                >1651
DCDF: 3F 45 58
DCE2: 54 52 41
DCE5: 20 49 47
  E8: 4E 4F 52
  EB: 45 44     >1652 EXIG      ASC  '?EXTRA IGNORED'
DCED: 0D 00     >1653           HEX  0D00
                >1654
DCEF: 3F 52 45
DCF2: 45 4E 54
DCF5: 45 52     >1655 REENT     ASC  '?REENTER'
DCF7: 0D 00     >1656           HEX  0D00
                >1657
DCF9: D0 04     >1658 NEXT      BNE  VARNXT     ;Branch if var specified
DCFB: A0 00     >1659           LDY  #0
DCFD: F0 03     >1660           BEQ  SKPV
DCFF: 20 E3 DF  >1661 VARNXT    JSR  PTRGET     ;Find var pointer
DD02: 85 85     >1662 SKPV      STA  FORPNT
DD04: 84 86     >1663           STY  FORPNT+1
DD06: 20 65 D3  >1664           JSR  GTFORPNT   ;Find FOR data on stack
DD09: F0 04     >1665           BEQ  GOTFOR
DD0B: A2 00     >1666           LDX  #NXwoFOR-ERRMSG
DD0D: F0 69     >1667 GERR      BEQ  JERROR     ;Always
DD0F: 9A        >1668 GOTFOR    TXS             ;Set stack ptr to point
DD10: E8        >1670           INX
DD11: E8        >1670           INX
DD12: E8        >1670           INX
DD13: E8        >1670           INX
DD14: 8A        >1672           TXA             ;Low byte of adrs of STEP value
DD15: E8        >1674           INX
  16: E8        >1674           INX
DD17: E8        >1674           INX
```

II

```
DD18: E8        >1674           INX
DD19: E8        >1674           INX
DD1A: E8        >1674           INX
DD1B: 86 60     >1676           STX  DEST       ;Low byte adrs of FOR var value
DD1D: A0 01     >1677           LDY  #1
DD1F: 20 F9 EA  >1678           JSR  MOVFM      ;STEP to FAC
DD22: BA        >1679           TSX
DD23: BD 09 01  >1680           LDA  STACK+9,X
DD26: 85 A2     >1681           STA  FACSGN
DD28: A5 85     >1682           LDA  FORPNT
DD2A: A4 86     >1683           LDY  FORPNT+1
DD2C: 20 BE E7  >1684           JSR  FADD       ;Add to FOR value
DD2F: 20 27 EB  >1685           JSR  SETFOR     ;Put new value back
DD32: A0 01     >1686           LDY  #1
DD34: 20 B4 EB  >1687           JSR  FCOMP2     ;Compare to end value
DD37: BA        >1688           TSX
DD38: 38        >1689           SEC
DD39: FD 09 01  >1690           SBC  STACK+9,X
DD3C: F0 17     >1691           BEQ  ENDFOR     ;Branch if FOR complete
DD3E: BD 0F 01  >1692           LDA  STACK+$F,X ;Otherwise set up
DD41: 85 75     >1693           STA  CURLIN     ; FOR line #
DD43: BD 10 01  >1694           LDA  STACK+$10,X
DD46: 85 76     >1695           STA  CURLIN+1
DD48: BD 12 01  >1696           LDA  STACK+$12,X ; and set TXTPTR to just
DD4B: 85 B8     >1697           STA  TXTPTR     ; after FOR statement
DD4D: BD 11 01  >1698           LDA  STACK+$11,X
DD50: 85 B9     >1699           STA  TXTPTR+1
DD52: 4C D2 D7  >1700 GONEWST   JMP  NEWSTT
DD55: 8A        >1701 ENDFOR    TXA
DD56: 69 11     >1702           ADC  #$11       ;Carry is set
DD58: AA        >1703           TAX             ;Cancel FOR by bumping
DD59: 9A        >1704           TXS             ; stack pointer by $12.
DD5A: 20 B7 00  >1705           JSR  CHRGOT
DD5D: C9 2C     >1706           CMP  #','       ;Another var in NEXT?
DD5F: D0 F1     >1707           BNE  GONEWST
DD61: 20 B1 00  >1708           JSR  CHRGET
DD64: 20 FF DC  >1709           JSR  VARNXT     ;Does not return
                >1710
DD67: 20 7B DD  >1711 FRMNUM    JSR  FRMEVL
DD6A: 18        >1712 CHKNUM    CLC
DD6B: 24        >1713           HEX  24         ;Dummy for skip
DD6C: 38        >1714 CHKSTR    SEC
DD6D: 24 11     >1715 CHKVAL    BIT  VALTYP
DD6F: 30 03     >1716           BMI  CV2
DD71: B0 03     >1717           BCS  MISMTCH
DD73: 60        >1718 RET12     RTS
DD74: B0 FD     >1719 CV2       BCS  RET12
DD76: A2 A3     >1720 MISMTCH   LDX  #TYPEMISS-ERRMSG
DD78: 4C 12 D4  >1721 JERROR    JMP  ERROR
                >1722
                 151            PUT  APSOFT II
                >1     ********************************
                >2     *                              *
                >3     *    Applesoft - Part II       *
                >4     *                              *
                >5     * Copywrite Apple Computer,  *
                >6     * Inc. and Microsoft, Inc.;  *
```

```
                >7    * not for publication or    *
                >8    * distribution.             *
                >9    *                           *
                >10   *****************************
                >11   *                           *
                >12   *   Formula evaluation,     *
                >13   *   Pointer locating,       *
                >14   *   & String Handling.      *
                >15   *                           *
                >16   *       $DD7B - $E79F       *
                >17   *                           *
                >18   *****************************
                >19
                >20   * Main formula evalutation routine.
                >21   * On entry TXTPTR points at 1st chr of formula.
                >22
                >23   FRMEVL    >>>  DECR.TXTPTR
DD7B: A6 B8     >23              LDX  TXTPTR
DD7D: D0 02     >23              BNE  ND
DD7F: C6 B9     >23              DEC  TXTPTR+1
DD81: C6 B8     >23   ND         DEC  TXTPTR
                >23              <<<
DD83: A2 00     >24              LDX  #0
DD85: 24        >25              DFB  $24          ;Trick to skip
DD86: 48        >26   FEVLOOP    PHA
DD87: 8A        >27              TXA
DD88: 48        >28              PHA
DD89: A9 01     >29              LDA  #1
DD8B: 20 D6 D3  >30              JSR  CHKMEM
DD8E: 20 60 DE  >31              JSR  GETVAL
DD91: A9 00     >32              LDA  #0
DD93: 85 89     >33              STA  CPRTYP
DD95: 20 B7 00  >34   FRMEVL2    JSR  CHRGOT
DD98: 38        >35   CPROP      SEC
DD99: E9 CF     >36              SBC  #$CF         ;> token
DD9B: 90 17     >37              BCC  CHKTYP
DD9D: C9 03     >38              CMP  #3           ;or =, <
DD9F: B0 13     >39              BGE  CHKTYP
DDA1: C9 01     >40              CMP  #1
DDA3: 2A        >41              ROL
DDA4: 49 01     >42              EOR  #1
DDA6: 45 89     >43              EOR  CPRTYP       ;Set bits of CPRTYP:
DDA8: C5 89     >44              CMP  CPRTYP       ; 00000>=<
DDAA: 90 61     >45              BCC  SNTXERR
DDAC: 85 89     >46              STA  CPRTYP
DDAE: 20 B1 00  >47              JSR  CHRGET       ;Another operator?
DDB1: 4C 98 DD  >48              JMP  CPROP        ;Check for <,=,> again
DDB4: A6 89     >49   CHKTYP     LDX  CPRTYP
DDB6: D0 2C     >50              BNE  COMPARE      ;Branch if had <,=,>
DDB8: B0 7B     >51              BCS  NOTMATH      ;Branch if next token > "<"
DDBA: 69 07     >52              ADC  #$CF-plus
DDBC: 90 77     >53              BCC  NOTMATH      ;Branch if next token < "+"
DDBE: 65 11     >54              ADC  VALTYP       ;+ and last result a string?
DDC0: D0 03     >55              BNE  ARITH        ;Branch if not
DDC2: 4C 97 E5  >56              JMP  CAT          ;Concatenate if so.
DDC5: 69 FF     >57   ARITH      ADC  #$FF         ;Now A-reg has offset
DDC7: 85 5E     >58              STA  INDEX        ; from "+".
```

```
DDC9: 0A        >59              ASL
DDCA: 65 5E     >60              ADC  INDEX       ;Times 3
DDCC: A8        >61              TAY
DDCD: 68        >62   PREFTEST   PLA              ;Get last preferance
  CE: D9 B2 D0  >63              CMP  MATHTBL,Y
DDD1: B0 67     >64              BGE  DOMTH       ;Do now if preferred
DDD3: 20 6A DD  >65              JSR  CHKNUM      ;Was last result a #?
DDD6: 48        >66   NXOP      PHA
DDD7: 20 FD DD  >67   SAVOP     JSR  PSHMAD      ;Save operation on stack
DDDA: 68        >68              PLA
DDDB: A4 87     >69              LDY  LASTOP
DDDD: 10 17     >70              BPL  PREFNC
DDDF: AA        >71              TAX
DDE0: F0 56     >72              BEQ  GOEX        ;Exit if no math in frm
DDE2: D0 5F     >73              BNE  DOMATH
                >74
DDE4: 46 11     >75   COMPARE   LSR  VALTYP      ;Enable string compare
DDE6: 8A        >76              TXA              ;Set CPRTYP: 0000>=<C
DDE7: 2A        >77              ROL              ; where C=carry from
                >78              >>>  DECR.TXTPTR
DDE8: A6 B8     >78              LDX  TXTPTR
DDEA: D0 02     >78              BNE  ND
DDEC: C6 B9     >78              DEC  TXTPTR+1
DDEE: C6 B8     >78   ND         DEC  TXTPTR
                >78              <<<
DDF0: A0 1B     >79              LDY  #PLUS-MATHTBL
DDF2: 85 89     >80              STA  CPRTYP      ; next character test.
DDF4: D0 D7     >81              BNE  PREFTEST    ;Always
                >82
 F6: D9 B2 D0   >83   PREFNC     CMP  MATHTBL,Y
 F9: B0 48      >84              BGE  DOMATH
DDFB: 90 D9     >85              BLT  NXOP
                >86
DDFD: B9 B4 D0  >87   PSHMAD    LDA  MATHTBL+2,Y
DE00: 48        >88              PHA              ;Put address of math
DE01: B9 B3 D0  >89              LDA  MATHTBL+1,Y
DE04: 48        >90              PHA              ;routine on stack.
DE05: 20 10 DE  >91              JSR  PSHF        ;Returns via the JMP (INDEX)
DE08: A5 89     >92              LDA  CPRTYP
DE0A: 4C 86 DD  >93              JMP  FEVLOOP
                >94
DE0D: 4C C9 DE  >95   SNTXERR   JMP  SYNERR
                >96
DE10: A5 A2     >97   PSHF      LDA  FACSGN      ;Get FACSGN to push it
DE12: BE B2 D0  >98              LDX  MATHTBL,Y
                >99
                >100  * Set INDEX for return and push FAC
                >101  * A-reg has FACSGN, or -1,0,1 if from STEP
                >102
DE15: A8        >103  PSHFACX   TAY              ;Called by STEP
DE16: 68        >104             PLA              ;Pull return address
DE17: 85 5E     >105             STA  INDEX       ;Place it in INDEX.
DE19: E6 5E     >106             INC  INDEX       ;This routine assumes
DE1B: 68        >107             PLA              ; return address is not
DE1C: 85 5F     >108             STA  INDEX+1     ; on page boundary.
  1E: 98        >109             TYA
DE1F: 48        >110             PHA              ;Push FACSGN
```

```
DE20: 20 72 EB >111  PUSHFAC  JSR  RNDB        ;Called by FOR
               >112           >>>  PUSH.FAC+3
DE23: A5 A1    >112           LDA  FAC+3+1
DE25: 48       >112           PHA
DE26: A5 A0    >112           LDA  FAC+3
DE28: 48       >112           PHA
               >112           <<<
               >113           >>>  PUSH.FAC+1
DE29: A5 9F    >113           LDA  FAC+1+1
DE2B: 48       >113           PHA
DE2C: A5 9E    >113           LDA  FAC+1
DE2E: 48       >113           PHA
               >113           <<<
DE2F: A5 9D    >114           LDA  FAC
DE31: 48       >115           PHA
DE32: 6C 5E 00 >116           JMP  (INDEX)     ;Do RTS funny way
               >117
DE35: A0 FF    >118  NOTMATH  LDY  #$FF        ;Set up to exit routine
DE37: 68       >119           PLA
DE38: F0 23    >120  GOEX     BEQ  EXIT        ;Exit if no math to do
DE3A: C9 64    >121  DOMTH    CMP  #$64        ;Was it "<"?
DE3C: F0 03    >122           BEQ  DMTH        ;Allow string compare if so
DE3E: 20 6A DD >123           JSR  CHKNUM
DE41: 84 87    >124  DMTH     STY  LASTOP
               >125
               >126  * Pull floating # from stack, place in ARG and go to
               >127  * math routine via RTS (address was placed on stack):
               >128
DE43: 68       >129  DOMATH   PLA
DE44: 4A       >130           LSR
DE45: 85 16    >131           STA  CPRMASK     ; 00000>=<
               >132           >>>  PULL.ARG
DE47: 68       >132           PLA
DE48: 85 A5    >132           STA  ARG
DE4A: 68       >132           PLA
DE4B: 85 A6    >132           STA  ARG+1
               >132           <<<
               >133           >>>  PULL.ARG+2
DE4D: 68       >133           PLA
DE4E: 85 A7    >133           STA  ARG+2
DE50: 68       >133           PLA
DE51: 85 A8    >133           STA  ARG+2+1
               >133           <<<
               >134           >>>  PULL.ARG+4
DE53: 68       >134           PLA
DE54: 85 A9    >134           STA  ARG+4
DE56: 68       >134           PLA
DE57: 85 AA    >134           STA  ARG+4+1
               >134           <<<
DE59: 45 A2    >135           EOR  FACSGN
DE5B: 85 AB    >136           STA  SGNCPR
DE5D: A5 9D    >137  EXIT     LDA  FAC
DE5F: 60       >138           RTS
               >139
               >140  * Get value of variable or number at TXTPNT, or point
               >141  * to string descriptor if a string, and put in FAC.
               >142
```

```
DE60: A9 00     >143  GETVAL   LDA  #0
DE62: 85 11     >144          STA  VALTYP      ; clear valtyp to "number"  (i.e. a quote or a var. name.
DE64: 20 B1 00  >145  SKIP    JSR  CHRGET      ; increment txtptr
DE67: B0 03     >146          BCS  VAR?        ; carry set if not an ascii number. carry clear if it is.
   69: 4C 4A EC >147  NUMBER  JMP  FIN         ;If numeric
DE6C: 20 7D E0  >148  VAR?    JSR  ISLETC      ;A variable? ie: a letter?
DE6F: B0 64     >149          BCS  VARL        ;carry set if a letter (ie a name)
DE71: C9 2E     >150          CMP  #'.'
DE73: F0 F4     >151          BEQ  NUMBER
DE75: C9 C9     >152          CMP  #minus
DE77: F0 55     >153          BEQ  MIN
DE79: C9 C8     >154          CMP  #plus       ; go back for next character
DE7B: F0 E7     >155          BEQ  SKIP
DE7D: C9 22     >156          CMP  #'"'
DE7F: D0 0F     >157          BNE  NOT?
DE81: A5 B8     >158  STRTXT   LDA  TXTPTR      ;Explicit string, build desc.
DE83: A4 B9     >159          LDY  TXTPTR+1
DE85: 69 00     >160          ADC  #0
DE87: 90 01     >161          BCC  ST1
DE89: C8        >162          INY
DE8A: 20 E7 E3  >163  ST1     JSR  STRLIT
DE8D: 4C 3D E7  >164          JMP  POINT       ;Get pointer to desc. in FAC
                >165
DE90: C9 C6     >166  NOT?    CMP  #not
DE92: D0 10     >167          BNE  FN?
DE94: A0 18     >168          LDY  #EQUAL-MATHTBL
DE96: D0 38     >169          BNE  EQUL        ;Always
                >170
DE98: A5 9D     >171  EQUOP   LDA  FAC
  9A: D0 03     >172          BNE  NOTZ
  9C: A0 01     >173          LDY  #1
DE9E: 2C        >174          HEX  2C          ;Trick to skip
DE9F: A0 00     >175  NOTZ    LDY  #0
DEA1: 4C 01 E3  >176          JMP  SNGFLT
                >177
DEA4: C9 C2     >178  FN?     CMP  #fn
DEA6: D0 03     >179          BNE  SGN?
DEA8: 4C 54 E3  >180          JMP  FUNCT
                >181
DEAB: C9 D2     >182  SGN?    CMP  #sgn
DEAD: 90 03     >183          BLT  PARCHK
DEAF: 4C 0C DF  >184          JMP  UNARY
                >185
DEB2: 20 BB DE  >186  PARCHK  JSR  CHKOPN      ;Is there a '(' at TXTPTR?
DEB5: 20 7B DD  >187          JSR  FRMEVL      ;If so, evaluate and
DEB8: A9 29     >188  CHKCLS  LDA  #')'        ;Check for ')'
DEBA: 2C        >189          HEX  2C          ;Trick
DEBB: A9 28     >190  CHKOPN  LDA  #'('
DEBD: 2C        >191          HEX  2C          ;Trick
DEBE: A9 2C     >192  CHKCOM  LDA  #','        ;Comma at TXTPTR?
DEC0: A0 00     >193  SYNCHR  LDY  #0
DEC2: D1 B8     >194          CMP  (TXTPTR),Y
DEC4: D0 03     >195          BNE  SYNERR
DEC6: 4C B1 00  >196          JMP  CHRGET      ;If ok, get next chr & return
                >197
  C9: A2 10     >198  SYNERR  LDX  #SYNTXERR-ERRMSG
DECB: 4C 12 D4  >199          JMP  ERROR
```

```
                    >200
DECE: A0 15         >201  MIN       LDY  #MINUS-MATHTBL
DED0: 68            >202  EQUL      PLA
DED1: 68            >203            PLA
 D2: 4C D7 DD       >204            JMP  SAVOP
                    >205
DED5: 20 E3 DF      >206  VARL      JSR  PTRGET      ; ~~is this a literal~~ variable name expected
DED8: 85 A0         >207            STA  VPNT
DEDA: 84 A1         >208            STY  VPNT+1
DEDC: A6 11         >209            LDX  VALTYP      ;String?
DEDE: F0 05         >210            BEQ  VR1         ;Branch if not
DEE0: A2 00         >211            LDX  #0
DEE2: 86 AC         >212            STX  STRNG1+1
DEE4: 60            >213            RTS
DEE5: A6 12         >214  VR1       LDX  VALTYP+1    ;Integer var?
DEE7: 10 0D         >215            BPL  VR2         ;Branch if not
DEE9: A0 00         >216            LDY  #0
DEEB: B1 A0         >217            LDA  (VPNT),Y
DEED: AA            >218            TAX
DEEE: C8            >219            INY
DEEF: B1 A0         >220            LDA  (VPNT),Y
DEF1: A8            >221            TAY
DEF2: 8A            >222            TXA
DEF3: 4C F2 E2      >223            JMP  GIVAYF
DEF6: 4C F9 EA      >224  VR2       JMP  MOVFM
                    >225
DEF9: 20 B1 00      >226  SCREEN    JSR  CHRGET
DEFC: 20 EC F1      >227            JSR  PLOTFNS
DEFF: 8A            >228            TXA
F00: A4 F0          >229            LDY  FIRST
 02: 20 71 F8       >230            JSR  SCRN
DF05: A8            >231            TAY
DF06: 20 01 E3      >232            JSR  SNGFLT
DF09: 4C B8 DE      >233            JMP  CHKCLS
                    >234
                    >235  * Process unary operators (functions):
                    >236
DF0C: C9 D7         >237  UNARY     CMP  #scrn       ;Not unary, do special
DF0E: F0 E9         >238            BEQ  SCREEN
DF10: 0A            >239            ASL
DF11: 48            >240            PHA
DF12: AA            >241            TAX
DF13: 20 B1 00      >242            JSR  CHRGET
DF16: E0 CF         >243            CPX  #leftstr*2-1
DF18: 90 20         >244            BLT  NOTinstr    ;Branch if not an instring op
DF1A: 20 BB DE      >245            JSR  CHKOPN      ;Check for '('
DF1D: 20 7B DD      >246            JSR  FRMEVL      ;Process concat., etc.
DF20: 20 BE DE      >247            JSR  CHKCOM
DF23: 20 6C DD      >248            JSR  CHKSTR      ;Make sure it is a string
DF26: 68            >249            PLA
DF27: AA            >250            TAX              ;Retrieve routine pointer
                    >251            >>>  PUSH.VPNT
DF28: A5 A1         >251            LDA  VPNT+1
DF2A: 48            >251            PHA
DF2B: A5 A0         >251            LDA  VPNT
 2D: 48             >251            PHA
                    >251            <<<
```

```
DF2E: 8A        >252            TXA
DF2F: 48        >253            PHA                 ;Push it back
DF30: 20 F8 E6  >254            JSR  GETBYT
DF33: 68        >255            PLA
DF34: A8        >256            TAY                 ;Point Y to routine
DF35: 8A        >257            TXA
DF36: 48        >258            PHA                 ;Push 1st param
DF37: 4C 3F DF  >259            JMP  GOROUT
                >260
DF3A: 20 B2 DE  >261  NOTinstr  JSR  PARCHK
DF3D: 68        >262            PLA
DF3E: A8        >263            TAY
DF3F: B9 DC CF  >264  GOROUT    LDA  UNFNC-$A4,Y ;$A4=sgn*2
DF42: 85 91     >265            STA  JMPADRS+1
DF44: B9 DD CF  >266            LDA  UNFNC-$A3,Y
DF47: 85 92     >267            STA  JMPADRS+2
DF49: 20 90 00  >268            JSR  JMPADRS        ;Does not return for
                >269                                ; LEFT$, RIGHT$, MID$.
DF4C: 4C 6A DD  >270            JMP  CHKNUM
                >271
DF4F: A5 A5     >272  OR        LDA  ARG
DF51: 05 9D     >273            ORA  FAC
DF53: D0 0B     >274            BNE  TRUE
DF55: A5 A5     >275  AND       LDA  ARG
DF57: F0 04     >276            BEQ  FALSE
DF59: A5 9D     >277            LDA  FAC
DF5B: D0 03     >278            BNE  TRUE
DF5D: A0 00     >279  FALSE     LDY  #0
DF5F: 2C        >280            HEX  2C             ;Trick
DF60: A0 01     >281  TRUE      LDY  #1
DF62: 4C 01 E3  >282            JMP  SNGFLT
                >283
DF65: 20 6D DD  >284  POSOP     JSR  CHKVAL
DF68: B0 13     >285            BCS  STRCMP         ;Branch if strings
DF6A: A5 AA     >286            LDA  ARGSGN
DF6C: 09 7F     >287            ORA  #$7F
DF6E: 25 A6     >288            AND  ARG+1
DF70: 85 A6     >289            STA  ARG+1
DF72: A9 A5     >290            LDA  #ARG
DF74: A0 00     >291            LDY  #0
DF76: 20 B2 EB  >292            JSR  FCOMP          ;Return A-reg = -1,0,1
DF79: AA        >293            TAX                 ; as ARG <,=,> FAC
DF7A: 4C B0 DF  >294            JMP  NUMCMP
DF7D: A9 00     >295  STRCMP    LDA  #0
DF7F: 85 11     >296            STA  VALTYP
DF81: C6 89     >297            DEC  CPRTYP         ;?
DF83: 20 00 E6  >298            JSR  FREFAC
DF86: 85 9D     >299            STA  FAC            ;String length
DF88: 86 9E     >300            STX  FAC+1
DF8A: 84 9F     >301            STY  FAC+2
DF8C: A5 A8     >302            LDA  ARG+3
DF8E: A4 A9     >303            LDY  ARG+4
DF90: 20 04 E6  >304            JSR  FRETMP
DF93: 86 A8     >305            STX  ARG+3
DF95: 84 A9     >306            STY  ARG+4
DF97: AA        >307            TAX                 ;Len (ARG) string
DF98: 38        >308            SEC
```

```
DF99: E5 9D     >309            SBC  FAC         ;Set X to smaller len
DF9B: F0 08     >310            BEQ  SFS
DF9D: A9 01     >311            LDA  #1
DF9F: 90 04     >312            BCC  SFS
  A1: A6 9D     >313            LDX  FAC
DFA3: A9 FF     >314            LDA  #$FF
DFA5: 85 A2     >315  SFS       STA  FACSGN      ;Flag which shorter
DFA7: A0 FF     >316            LDY  #$FF
DFA9: E8        >317            INX
DFAA: C8        >318  CMPLOOP   INY
DFAB: CA        >319            DEX
DFAC: D0 07     >320            BNE  DOCMP
DFAE: A6 A2     >321            LDX  FACSGN      ;If = so far, decide by len
DFB0: 30 0F     >322  NUMCMP    BMI  CMPDONE
DFB2: 18        >323            CLC
DFB3: 90 0C     >324            BCC  CMPDONE
DFB5: B1 A8     >325  DOCMP     LDA  (ARG+3),Y
DFB7: D1 9E     >326            CMP  (FAC+1),Y
DFB9: F0 EF     >327            BEQ  CMPLOOP
DFBB: A2 FF     >328            LDX  #$FF
DFBD: B0 02     >329            BCS  CMPDONE
DFBF: A2 01     >330            LDX  #1
DFC1: E8        >331  CMPDONE   INX              ;Convert FF,0,1 to 1,2,4
DFC2: 8A        >332            TXA
DFC3: 2A        >333            ROL
DFC4: 25 16     >334            AND  CPRMASK     ; 00000>=<
DFC6: F0 02     >335            BEQ  JF          ;If no match: false
DFC8: A9 01     >336            LDA  #1          ;At least one match: true
DFCA: 4C 93 EB  >337  JF        JMP  FLOAT
                >338
..CD: 20 FB E6  >339  PDL       JSR  CONINT      ;Get # in X (<4 not checked)
DFD0: 20 1E FB  >340            JSR  PREAD       ;Read paddle
DFD3: 4C 01 E3  >341            JMP  SNGFLT      ;Float result
                >342
DFD6: 20 BE DE  >343  NXDIM     JSR  CHKCOM
DFD9: AA        >344  DIM       TAX
DFDA: 20 E8 DF  >345            JSR  PTRGET2
DFDD: 20 B7 00  >346            JSR  CHRGOT
DFE0: D0 F4     >347            BNE  NXDIM
DFE2: 60        >348            RTS
                >349
DFE3: A2 00     >350  PTRGET    LDX  #0
DFE5: 20 B7 00  >351            JSR  CHRGOT      ;Get variable name
DFE8: 86 10     >352  PTRGET2   STX  DIMFLG      ;X has VARNAM if from DIM
DFEA: 85 81     >353  PTRGET3   STA  VARNAM
DFEC: 20 B7 00  >354            JSR  CHRGOT
DFEF: 20 7D E0  >355            JSR  ISLETC      ;Is it a letter?
DFF2: B0 03     >356            BCS  NAMOK       ;Branch if so
DFF4: 4C C9 DE  >357  BADNAM    JMP  SYNERR      ;Error if not
DFF7: A2 00     >358  NAMOK     LDX  #0
DFF9: 86 11     >359            STX  VALTYP
DFFB: 86 12     >360            STX  VALTYP+1
DFFD: 4C 07 E0  >361            JMP  MORNAM      ;To branch across $E000 vectors
                >362
                >363  * BASIC entry points for DOS etc., use:
                >364
E000: 4C 28 F1  >365            JMP  COLDST
```

```
E003: 4C 3C D4 >366           JMP  RESTART
E006: 00       >367           BRK
               >368
E007: 20 B1 00 >369  MORNAM    JSR  CHRGET      ;2nd chr of variable name
E00A: 90 05    >370           BCC  GTLT        ;Branch if numeric
E00C: 20 7D E0 >371           JSR  ISLETC      ;Is it alpha?
E00F: 90 0B    >372           BCC  STRNG?      ;Branch if not
E011: AA       >373  GTLT     TAX              ;Save 2nd chr of name in X
E012: 20 B1 00 >374  BYPASS   JSR  CHRGET      ;Find end of var. name
E015: 90 FB    >375           BCC  BYPASS      ;Loop if numeric
E017: 20 7D E0 >376           JSR  ISLETC
E01A: B0 F6    >377           BCS  BYPASS      ;or if alpha.
E01C: C9 24    >378  STRNG?   CMP  #'$'        ;Set up var type flags
E01E: D0 06    >379           BNE  INTVAR?
E020: A9 FF    >380           LDA  #$FF
E022: 85 11    >381           STA  VALTYP
E024: D0 10    >382           BNE  NIN
E026: C9 25    >383  INTVAR?  CMP  #'%'
E028: D0 13    >384           BNE  SCDCH
E02A: A5 14    >385           LDA  SUBFLG      ;Integer var allowed?
E02C: 30 C6    >386           BMI  BADNAM      ;Error if not.
E02E: A9 80    >387           LDA  #$80
E030: 85 12    >388           STA  VALTYP+1
E032: 05 81    >389           ORA  VARNAM
E034: 85 81    >390           STA  VARNAM
E036: 8A       >391  NIN      TXA
E037: 09 80    >392           ORA  #$80
E039: AA       >393           TAX
E03A: 20 B1 00 >394           JSR  CHRGET
E03D: 86 82    >395  SCDCH    STX  VARNAM+1    ;2nd var name chr
E03F: 38       >396           SEC
E040: 05 14    >397           ORA  SUBFLG
E042: E9 28    >398           SBC  #'('
E044: D0 03    >399           BNE  BSB
E046: 4C 1E E1 >400  JARY     JMP  ARRAY
E049: 24 14    >401  BSB      BIT  SUBFLG
E04B: 30 02    >402           BMI  VSEARCH
E04D: 70 F7    >403           BVS  JARY        ;Branch if called by GETARYPT
E04F: A9 00    >404  VSEARCH  LDA  #0
E051: 85 14    >405           STA  SUBFLG
E053: A5 69    >406           LDA  VARTAB      ;Init varl pntr
E055: A6 6A    >407           LDX  VARTAB+1
E057: A0 00    >408           LDY  #0
E059: 86 9C    >409  NXVAR    STX  LOWTR+1
E05B: 85 9B    >410  NV1      STA  LOWTR
E05D: E4 6C    >411           CPX  ARYTAB+1    ;End of simple vars?
E05F: D0 04    >412           BNE  NV2         ;No, go on
E061: C5 6B    >413           CMP  ARYTAB
E063: F0 22    >414           BEQ  NOTFND      ;Yes, make one?
E065: A5 81    >415  NV2      LDA  VARNAM
E067: D1 9B    >416           CMP  (LOWTR),Y
E069: D0 08    >417           BNE  NXPTR       ;Branch if not this one
E06B: A5 82    >418           LDA  VARNAM+1
E06D: C8       >419           INY
E06E: D1 9B    >420           CMP  (LOWTR),Y
E070: F0 6C    >421           BEQ  SETVPNT     ;Branch if found
E072: 88       >422           DEY
```

```
E073: 18        >423  NXPTR    CLC
E074: A5 9B     >424          LDA  LOWTR
E076: 69 07     >425          ADC  #7
E078: 90 E1     >426          BCC  NV1
 7A: E8         >427          INX
E07B: D0 DC     >428          BNE  NXVAR      ;Always
                >429
                >430  * Check if letter A-Z, set carry if so, clear if not:
                >431
E07D: C9 41     >432  ISLETC   CMP  #'A'
E07F: 90 05     >433          BLT  RTN1
E081: E9 5B     >434          SBC  #'[
E083: 38        >435          SEC
E084: E9 A5     >436          SBC  #$100-'[   ;Get orig. A-reg
E086: 60        >437  RTN1     RTS
                >438
E087: 68        >439  NOTFND   PLA             ;If called by VARL it
E088: 48        >440          PHA             ; is not an assignment,
E089: C9 D7     >441          CMP  #VARL+2    ; so fake it.
E08B: D0 0F     >442          BNE  NEWVAR
E08D: BA        >443          TSX
E08E: BD 02 01  >444          LDA  STACK+2,X
E091: C9 DE     >445          CMP  #>VARL+2
E093: D0 07     >446          BNE  NEWVAR
E095: A9 9A     >447          LDA  #TWOBRK
E097: A0 E0     >448          LDY  #>TWOBRK
E099: 60        >449          RTS
E09A: 00        >450  TWOBRK   BRK
E09B: 00        >451          BRK
                >452
                >453  * Move arrays to make room for new variable:
                >454
                >455  NEWVAR    >>>  TRAY.ARYTAB;LOWTR
E09C: A5 6B     >455          LDA  ARYTAB
E09E: A4 6C     >455          LDY  ARYTAB+1
E0A0: 85 9B     >455          STA  LOWTR
E0A2: 84 9C     >455          STY  LOWTR+1
                >455          <<<
                >456          >>>  TRAY.STREND;HIGHTR
E0A4: A5 6D     >456          LDA  STREND
E0A6: A4 6E     >456          LDY  STREND+1
E0A8: 85 96     >456          STA  HIGHTR
E0AA: 84 97     >456          STY  HIGHTR+1
                >456          <<<
E0AC: 18        >457          CLC
E0AD: 69 07     >458          ADC  #7
E0AF: 90 01     >459          BCC  NWV
E0B1: C8        >460          INY
E0B2: 85 94     >461  NWV      STA  ARYPNT
E0B4: 84 95     >462          STY  ARYPNT+1
E0B6: 20 93 D3  >463          JSR  BLTU
E0B9: A5 94     >464          LDA  ARYPNT
E0BB: A4 95     >465          LDY  ARYPNT+1
E0BD: C8        >466          INY
E0BE: 85 6B     >467          STA  ARYTAB
 C0: 84 6C      >468          STY  ARYTAB+1
E0C2: A0 00     >469          LDY  #0
```

```
E0C4: A5 81     >470            LDA  VARNAM
E0C6: 91 9B     >471            STA  (LOWTR),Y
E0C8: C8        >472            INY
E0C9: A5 82     >473            LDA  VARNAM+1
E0CB: 91 9B     >474            STA  (LOWTR),Y
E0CD: A9 00     >475            LDA  #0          ;Set value to 0
E0CF: C8        >477            INY
E0D0: 91 9B     >478            STA  (LOWTR),Y
E0D2: C8        >477            INY
E0D3: 91 9B     >478            STA  (LOWTR),Y
E0D5: C8        >477            INY
E0D6: 91 9B     >478            STA  (LOWTR),Y
E0D8: C8        >477            INY
E0D9: 91 9B     >478            STA  (LOWTR),Y
E0DB: C8        >477            INY
E0DC: 91 9B     >478            STA  (LOWTR),Y
E0DE: A5 9B     >480  SETVPNT   LDA  LOWTR
E0E0: 18        >481            CLC
E0E1: 69 02     >482            ADC  #2
E0E3: A4 9C     >483            LDY  LOWTR+1
E0E5: 90 01     >484            BCC  SVP
E0E7: C8        >485            INY
E0E8: 85 83     >486  SVP       STA  VARPNT
E0EA: 84 84     >487            STY  VARPNT+1
E0EC: 60        >488            RTS
                >489
E0ED: A5 0F     >490  GETARY    LDA  NUMDIM      ;Get # of dimensions
E0EF: 0A        >491  GETARY2   ASL              ; times 2
E0F0: 69 05     >492            ADC  #5          ; plus 5 (name, offset and #dim)
E0F2: 65 9B     >493            ADC  LOWTR       ;Add to variable pointer
E0F4: A4 9C     >494            LDY  LOWTR+1
E0F6: 90 01     >495            BCC  GD
E0F8: C8        >496            INY
E0F9: 85 94     >497  GD        STA  ARYPNT      ;Now points to first descriptor
E0FB: 84 95     >498            STY  ARYPNT+1    ; in array.
E0FD: 60        >499            RTS
                >500
E0FE: 90 80 00
E101: 00        >501  NEGNUM    HEX  90800000    ;=32768
                >502
E102: 20 B1 00  >503  MAKINT    JSR  CHRGET
E105: 20 67 DD  >504            JSR  FRMNUM
E108: A5 A2     >505  MKINT     LDA  FACSGN      ;Error if -
E10A: 30 0D     >506            BMI  MI1
E10C: A5 9D     >507  AYINT     LDA  FAC
E10E: C9 90     >508            CMP  #$90        ;<2^15 ?
E110: 90 09     >509            BCC  MI2         ;Branch if so
E112: A9 FE     >510            LDA  #NEGNUM     ;=2^15 ?
E114: A0 E0     >511            LDY  #>NEGNUM
E116: 20 B2 EB  >512            JSR  FCOMP
E119: D0 7E     >513  MI1       BNE  IQERR       ;Error if not
E11B: 4C F2 EB  >514  MI2       JMP  QINT
                >515
                >516  * Routine to locate array element or to create an array.
                >517
E11E: A5 14     >518  ARRAY     LDA  SUBFLG      ;Subscripts given?
E120: D0 47     >519            BNE  FNDARY      ;Branch if not
```

```
E122: A5 10     >520            LDA  DIMFLG
E124: 05 12     >521            ORA  VALTYP+1    ;Set high bit if %
E126: 48        >522            PHA
E127: A5 11     >523            LDA  VALTYP
 29: 48         >524            PHA
E12A: A0 00     >525            LDY  #0
E12C: 98        >526  NXTDIM    TYA
E12D: 48        >527            PHA
                >528            >>>  PUSH.VARNAM
E12E: A5 82     >528            LDA  VARNAM+1
E130: 48        >528            PHA
E131: A5 81     >528            LDA  VARNAM
E133: 48        >528            PHA
                >528            <<<
E134: 20 02 E1  >529            JSR  MAKINT
                >530            >>>  PULL.VARNAM
E137: 68        >530            PLA
E138: 85 81     >530            STA  VARNAM
E13A: 68        >530            PLA
E13B: 85 82     >530            STA  VARNAM+1
                >530            <<<
E13D: 68        >531            PLA
E13E: A8        >532            TAY
E13F: BA        >533            TSX
E140: BD 02 01  >534            LDA  STACK+2,X   ;Get rtn adrs
E143: 48        >535            PHA              ; and duplicate
E144: BD 01 01  >536            LDA  STACK+1,X
E147: 48        >537            PHA
E148: A5 A0     >538            LDA  FAC+3       ;Get subscript
 14A: 9D 02 01  >539            STA  STACK+2,X   ; and put on stack
 14D: A5 A1     >540            LDA  FAC+4       ; where rtn adrs was.
E14F: 9D 01 01  >541            STA  STACK+1,X
E152: C8        >542            INY
E153: 20 B7 00  >543            JSR  CHRGOT
E156: C9 2C     >544            CMP  #','
E158: F0 D2     >545            BEQ  NXTDIM
E15A: 84 0F     >546            STY  NUMDIM
E15C: 20 B8 DE  >547            JSR  CHKCLS
                >548            >>>  PULL.VALTYP
E15F: 68        >548            PLA
E160: 85 11     >548            STA  VALTYP
E162: 68        >548            PLA
E163: 85 12     >548            STA  VALTYP+1
                >548            <<<
E165: 29 7F     >549            AND  #$7F        ;Mask bit from VALTYP+1
E167: 85 10     >550            STA  DIMFLG
E169: A6 6B     >551  FNDARY    LDX  ARYTAB
E16B: A5 6C     >552            LDA  ARYTAB+1
E16D: 86 9B     >553  ARYLOOP   STX  LOWTR
E16F: 85 9C     >554            STA  LOWTR+1
E171: C5 6E     >555            CMP  STREND+1
E173: D0 04     >556            BNE  ARYNAM?
E175: E4 6D     >557            CPX  STREND
E177: F0 3F     >558            BEQ  NOTFOUND
E179: A0 00     >559  ARYNAM?   LDY  #0
 17B: B1 9B     >560            LDA  (LOWTR),Y
E17D: C8        >561            INY
```

```
E17E: C5 81     >562            CMP  VARNAM
E180: D0 06     >563            BNE  NXARY
E182: A5 82     >564            LDA  VARNAM+1
E184: D1 9B     >565            CMP  (LOWTR),Y
 86: F0 16      >566            BEQ  ARYFOUND
E188: C8        >567  NXARY     INY
E189: B1 9B     >568            LDA  (LOWTR),Y
E18B: 18        >569            CLC
E18C: 65 9B     >570            ADC  LOWTR
E18E: AA        >571            TAX
E18F: C8        >572            INY
E190: B1 9B     >573            LDA  (LOWTR),Y
E192: 65 9C     >574            ADC  LOWTR+1
E194: 90 D7     >575            BCC  ARYLOOP
E196: A2 6B     >576  SUBERR    LDX  #BADSUBS-ERRMSG
E198: 2C        >577            HEX  2C          ;Trick
E199: A2 35     >578  IQERR     LDX  #ILLQUAN-ERRMSG
E19B: 4C 12 D4  >579  JER       JMP  ERROR
                >580
E19E: A2 78     >581  ARYFOUND  LDX  #REdimARR-ERRMSG
E1A0: A5 10     >582            LDA  DIMFLG
E1A2: D0 F7     >583            BNE  JER
E1A4: A5 14     >584            LDA  SUBFLG
E1A6: F0 02     >585            BEQ  CHKDIM
E1A8: 38        >586            SEC
E1A9: 60        >587            RTS
                >588
E1AA: 20 ED E0  >589  CHKDIM    JSR  GETARY
E1AD: A5 0F     >590            LDA  NUMDIM
 AF: A0 04      >591            LDY  #4
 B1: D1 9B      >592            CMP  (LOWTR),Y
E1B3: D0 E1     >593            BNE  SUBERR
E1B5: 4C 4B E2  >594            JMP  FNDELEM
                >595
E1B8: A5 14     >596  NOTFOUND  LDA  SUBFLG
E1BA: F0 05     >597            BEQ  MAKARY
E1BC: A2 2A     >598            LDX  #0ofDATA-ERRMSG
E1BE: 4C 12 D4  >599            JMP  ERROR
                >600
E1C1: 20 ED E0  >601  MAKARY    JSR  GETARY
E1C4: 20 E3 D3  >602            JSR  REASON
E1C7: A9 00     >603            LDA  #0
E1C9: A8        >604            TAY
E1CA: 85 AE     >605            STA  STRNG2+1
E1CC: A2 05     >606            LDX  #5
E1CE: A5 81     >607            LDA  VARNAM
E1D0: 91 9B     >608            STA  (LOWTR),Y
E1D2: 10 01     >609            BPL  NINT
E1D4: CA        >610            DEX              ;Integer array
E1D5: C8        >611  NINT      INY
E1D6: A5 82     >612            LDA  VARNAM+1
E1D8: 91 9B     >613            STA  (LOWTR),Y
E1DA: 10 02     >614            BPL  RAR         ;Branch if real array
E1DC: CA        >615            DEX
E1DD: CA        >616            DEX
 DE: 86 AD      >617  RAR       STX  STRNG2      ;X=5,3,2 as: real,str,int
E1E0: A5 0F     >618            LDA  NUMDIM
```

```
E1E2: C8        >619            INY
E1E3: C8        >620            INY
E1E4: C8        >621            INY
E1E5: 91 9B     >622            STA  (LOWTR),Y
  E7: A2 0B     >623  SAVDIM    LDX  #11          ;Default dimension + 1
E1E9: A9 00     >624            LDA  #0
E1EB: 24 10     >625            BIT  DIMFLG       ;Dimensioned array?
E1ED: 50 08     >626            BVC  DFLTDIM      ;Branch if not
E1EF: 68        >627            PLA               ;Get specified dim in A,X
E1F0: 18        >628            CLC
E1F1: 69 01     >629            ADC  #1
E1F3: AA        >630            TAX
E1F4: 68        >631            PLA
E1F5: 69 00     >632            ADC  #0
E1F7: C8        >633  DFLTDIM   INY               ;Build dim table
E1F8: 91 9B     >634            STA  (LOWTR),Y
E1FA: C8        >635            INY
E1FB: 8A        >636            TXA
E1FC: 91 9B     >637            STA  (LOWTR),Y
E1FE: 20 AD E2  >638            JSR  MULT
E201: 86 AD     >639            STX  STRNG2
E203: 85 AE     >640            STA  STRNG2+1
E205: A4 5E     >641            LDY  INDEX        ;Retrieve Y saved by MULT
E207: C6 0F     >642            DEC  NUMDIM       ;Count down # dims
E209: D0 DC     >643            BNE  SAVDIM       ;Loop till done
E20B: 65 95     >644            ADC  ARYPNT+1
E20D: B0 5D     >645            BCS  GME
E20F: 85 95     >646            STA  ARYPNT+1
E211: A8        >647            TAY
 12: 8A         >648            TXA
 13: 65 94      >649            ADC  ARYPNT
E215: 90 03     >650            BCC  ZARY
E217: C8        >651            INY
E218: F0 52     >652            BEQ  GME
E21A: 20 E3 D3  >653  ZARY      JSR  REASON       ;Make sure there is room
E21D: 85 6D     >654            STA  STREND       ; and then zero out
E21F: 84 6E     >655            STY  STREND+1     ; the array.
E221: A9 00     >656            LDA  #0
E223: E6 AE     >657            INC  STRNG2+1
E225: A4 AD     >658            LDY  STRNG2
E227: F0 05     >659            BEQ  NXPG
E229: 88        >660  ZLUP      DEY
E22A: 91 94     >661            STA  (ARYPNT),Y
E22C: D0 FB     >662            BNE  ZLUP
E22E: C6 95     >663  NXPG      DEC  ARYPNT+1     ;Point to next page
E230: C6 AE     >664            DEC  STRNG2+1
E232: D0 F5     >665            BNE  ZLUP         ;Loop till done
E234: E6 95     >666            INC  ARYPNT+1
E236: 38        >667            SEC
E237: A5 6D     >668            LDA  STREND
E239: E5 9B     >669            SBC  LOWTR
E23B: A0 02     >670            LDY  #2
E23D: 91 9B     >671            STA  (LOWTR),Y
E23F: A5 6E     >672            LDA  STREND+1
E241: C8        >673            INY
  42: E5 9C     >674            SBC  LOWTR+1
E244: 91 9B     >675            STA  (LOWTR),Y
```

```
E246: A5 10     >676            LDA  DIMFLG
E248: D0 62     >677            BNE  RTN2
E24A: C8        >678            INY
E24B: B1 9B     >679  FNDELEM   LDA  (LOWTR),Y  ;Find specified element
  4D: 85 0F     >680            STA  NUMDIM     ; of array from index put
E24F: A9 00     >681            LDA  #0         ; on stack by NXTDIM.
E251: 85 AD     >682            STA  STRNG2
E253: 85 AE     >683  DIMLUP    STA  STRNG2+1
E255: C8        >684            INY
E256: 68        >685            PLA
E257: AA        >686            TAX
E258: 85 A0     >687            STA  FAC+3      ;Retrieve index and
E25A: 68        >688            PLA             ; check against dim.
E25B: 85 A1     >689            STA  FAC+4
E25D: D1 9B     >690            CMP  (LOWTR),Y
E25F: 90 0E     >691            BCC  DIMOK
E261: D0 06     >692            BNE  GSE
E263: C8        >693            INY
E264: 8A        >694            TXA
E265: D1 9B     >695            CMP  (LOWTR),Y
E267: 90 07     >696            BCC  DIMOK2
E269: 4C 96 E1  >697  GSE       JMP  SUBERR
                >698
E26C: 4C 10 D4  >699  GME       JMP  MEMERR
                >700
E26F: C8        >701  DIMOK     INY
E270: A5 AE     >702  DIMOK2    LDA  STRNG2+1
E272: 05 AD     >703            ORA  STRNG2
E274: 18        >704            CLC
E275: F0 0A     >705            BEQ  NXDM
  77: 20 AD E2  >706            JSR  MULT
E27A: 8A        >707            TXA
E27B: 65 A0     >708            ADC  FAC+3
E27D: AA        >709            TAX
E27E: 98        >710            TYA
E27F: A4 5E     >711            LDY  INDEX      ;Retrieve Y saved by MULT
E281: 65 A1     >712  NXDM      ADC  FAC+4      ;Next dim
E283: 86 AD     >713            STX  STRNG2
E285: C6 0F     >714            DEC  NUMDIM
E287: D0 CA     >715            BNE  DIMLUP     ;Loop till done
E289: 85 AE     >716            STA  STRNG2+1
E28B: A2 05     >717            LDX  #5
E28D: A5 81     >718            LDA  VARNAM
E28F: 10 01     >719            BPL  NINTA      ;Branch if not int
E291: CA        >720            DEX
E292: A5 82     >721  NINTA     LDA  VARNAM+1
E294: 10 02     >722            BPL  RARY       ;Branch if real
E296: CA        >723            DEX
E297: CA        >724            DEX
E298: 86 64     >725  RARY      STX  RESULT+2
E29A: A9 00     >726            LDA  #0
E29C: 20 B6 E2  >727            JSR  MU1
E29F: 8A        >728            TXA
E2A0: 65 94     >729            ADC  ARYPNT
E2A2: 85 83     >730            STA  VARPNT
  A4: 98        >731            TYA
E2A5: 65 95     >732            ADC  ARYPNT+1
```

```
E2A7: 85 84     >733            STA  VARPNT+1
E2A9: A8        >734            TAY
E2AA: A5 83     >735            LDA  VARPNT
E2AC: 60        >736  RTN2      RTS
                >737
                >738  * 16 bit (non floating) multiply of (LOWTR),Y
                >739  * by STRNG2 leaving product in A,X.
                >740  * Used only by array subscript routines.
                >741
E2AD: 84 5E     >742  MULT      STY  INDEX
E2AF: B1 9B     >743            LDA  (LOWTR),Y
E2B1: 85 64     >744            STA  RESULT+2
E2B3: 88        >745            DEY
E2B4: B1 9B     >746            LDA  (LOWTR),Y
E2B6: 85 65     >747  MU1       STA  RESULT+3
E2B8: A9 10     >748            LDA  #$10       ;Index for 16 bit mult
E2BA: 85 99     >749            STA  INDX
E2BC: A2 00     >750            LDX  #0
E2BE: A0 00     >751            LDY  #0
E2C0: 8A        >752  MU2       TXA
E2C1: 0A        >753            ASL
E2C2: AA        >754            TAX
E2C3: 98        >755            TYA
E2C4: 2A        >756            ROL
E2C5: A8        >757            TAY
E2C6: B0 A4     >758            BCS  GME
E2C8: 06 AD     >759            ASL  STRNG2
E2CA: 26 AE     >760            ROL  STRNG2+1
E2CC: 90 0B     >761            BCC  MU3
E2CE: 18        >762            CLC
E2CF: 8A        >763            TXA
E2D0: 65 64     >764            ADC  RESULT+2
E2D2: AA        >765            TAX
E2D3: 98        >766            TYA
E2D4: 65 65     >767            ADC  RESULT+3
E2D6: A8        >768            TAY
E2D7: B0 93     >769            BCS  GME
E2D9: C6 99     >770  MU3       DEC  INDX
E2DB: D0 E3     >771            BNE  MU2
E2DD: 60        >772            RTS
                >773
E2DE: A5 11     >774  FRE       LDA  VALTYP
E2E0: F0 03     >775            BEQ  FRE2
E2E2: 20 00 E6  >776            JSR  FREFAC
E2E5: 20 84 E4  >777  FRE2      JSR  GARBAG
E2E8: 38        >778            SEC
E2E9: A5 6F     >779            LDA  FRETOP
E2EB: E5 6D     >780            SBC  STREND
E2ED: A8        >781            TAY
E2EE: A5 70     >782            LDA  FRETOP+1
E2F0: E5 6E     >783            SBC  STREND+1
E2F2: A2 00     >784  GIVAYF    LDX  #0         ;Float signed integer in A,Y
E2F4: 86 11     >785            STX  VALTYP     ;Flag as number
E2F6: 85 9E     >786            STA  FAC+1
E2F8: 84 9F     >787            STY  FAC+2
E2FA: A2 90     >788            LDX  #$90
E2FC: 4C 9B EB  >789            JMP  FLO1
```

```
                        >790
E2FF: A4 24             >791  POS       LDY  CH
E301: A9 00             >792  SNGFLT    LDA  #0
E303: 38                >793            SEC
  04: F0 EC             >794            BEQ  GIVAYF
                        >795
E306: A6 76             >796  ERRDIR    LDX  CURLIN+1
E308: E8                >797            INX
E309: D0 A1             >798            BNE  RTN2        ;Return if deferred mode
E30B: A2 95             >799            LDX  #ILLDIR-ERRMSG
E30D: 2C                >800            HEX  2C          ;Trick
E30E: A2 E0             >801  UNDFNC    LDX  #UNDFUNC-ERRMSG
E310: 4C 12 D4          >802            JMP  ERROR
                        >803
E313: 20 41 E3          >804  DEF       JSR  FNC?
E316: 20 06 E3          >805            JSR  ERRDIR
E319: 20 BB DE          >806            JSR  CHKOPN
E31C: A9 80             >807            LDA  #$80
E31E: 85 14             >808            STA  SUBFLG      ;Disallow int vars, etc
E320: 20 E3 DF          >809            JSR  PTRGET      ;Get ptr to argument
E323: 20 6A DD          >810            JSR  CHKNUM
E326: 20 B8 DE          >811            JSR  CHKCLS
E329: A9 D0             >812            LDA  #equal
E32B: 20 C0 DE          >813            JSR  SYNCHR
E32E: 48                >814            PHA              ;1st chr follg =
                        >815            >>>  PUSH.VARPNT
E32F: A5 84             >815            LDA  VARPNT+1
E331: 48                >815            PHA
E332: A5 83             >815            LDA  VARPNT
  34: 48                >815            PHA
                        >815            <<<
                        >816            >>>  PUSH.TXTPTR
E335: A5 B9             >816            LDA  TXTPTR+1
E337: 48                >816            PHA
E338: A5 B8             >816            LDA  TXTPTR
E33A: 48                >816            PHA
                        >816            <<<
E33B: 20 95 D9          >817            JSR  DATA        ;Skip to next statement
E33E: 4C AF E3          >818            JMP  FNCDATA     ;Set up pointers in "value"
                        >819
                        >820  * For FNC, "value" of FNCNAM contains:
                        >821  *  Pointer to defn
                        >822  *  Pointer to argument variable
                        >823  *  1st chr of def
                        >824
E341: A9 C2             >825  FNC?      LDA  #fn
E343: 20 C0 DE          >826            JSR  SYNCHR
E346: 09 80             >827            ORA  #$80
E348: 85 14             >828            STA  SUBFLG
E34A: 20 EA DF          >829            JSR  PTRGET3
E34D: 85 8A             >830            STA  FNCNAM
E34F: 84 8B             >831            STY  FNCNAM+1
E351: 4C 6A DD          >832            JMP  CHKNUM
                        >833
E354: 20 41 E3          >834  FUNCT     JSR  FNC?
                        >835            >>>  PUSH.FNCNAM
E357: A5 8B             >835            LDA  FNCNAM+1
```

```
E359: 48          >835             PHA
E35A: A5 8A       >835             LDA  FNCNAM
E35C: 48          >835             PHA
                  >835             <<<
E35D: 20 B2 DE    >836             JSR  PARCHK
E360: 20 6A DD    >837             JSR  CHKNUM
                  >838             >>>  PULL.FNCNAM
E363: 68          >838             PLA
E364: 85 8A       >838             STA  FNCNAM
E366: 68          >838             PLA
E367: 85 8B       >838             STA  FNCNAM+1
                  >838             <<<
E369: A0 02       >839             LDY  #2
E36B: B1 8A       >840             LDA  (FNCNAM),Y
E36D: 85 83       >841             STA  VARPNT
E36F: AA          >842             TAX
E370: C8          >843             INY
E371: B1 8A       >844             LDA  (FNCNAM),Y
E373: F0 99       >845             BEQ  UNDFNC
E375: 85 84       >846             STA  VARPNT+1
E377: C8          >847             INY
E378: B1 83       >848  SAVOLD     LDA  (VARPNT),Y ;Save data
E37A: 48          >849             PHA
E37B: 88          >850             DEY
E37C: 10 FA       >851             BPL  SAVOLD
E37E: A4 84       >852             LDY  VARPNT+1   ;Get val of argument
E380: 20 2B EB    >853             JSR  MOVMF      ;FAC -> (VARPNT)
                  >854             >>>  PUSH.TXTPTR ;Remember position
E383: A5 B9       >854             LDA  TXTPTR+1
E385: 48          >854             PHA
E386: A5 B8       >854             LDA  TXTPTR
E388: 48          >854             PHA
                  >854             <<<
E389: B1 8A       >855             LDA  (FNCNAM),Y ;Y=0
E38B: 85 B8       >856             STA  TXTPTR     ;Point to fnc defn
E38D: C8          >857             INY
E38E: B1 8A       >858             LDA  (FNCNAM),Y
E390: 85 B9       >859             STA  TXTPTR+1
                  >860             >>>  PUSH.VARPNT
E392: A5 84       >860             LDA  VARPNT+1
E394: 48          >860             PHA
E395: A5 83       >860             LDA  VARPNT
E397: 48          >860             PHA
                  >860             <<<
E398: 20 67 DD    >861             JSR  FRMNUM     ;Evaluate fnc
                  >862             >>>  PULL.FNCNAM
E39B: 68          >862             PLA
E39C: 85 8A       >862             STA  FNCNAM
E39E: 68          >862             PLA
E39F: 85 8B       >862             STA  FNCNAM+1
                  >862             <<<
E3A1: 20 B7 00    >863             JSR  CHRGOT
E3A4: F0 03       >864             BEQ  GETOLD
E3A6: 4C C9 DE    >865             JMP  SYNERR
                  >866
                  >867  GETOLD     >>>  PULL.TXTPTR ;Retrieve prog position
E3A9: 68          >867             PLA
```

```
E3AA: 85 B8     >867            STA  TXTPTR
E3AC: 68        >867            PLA
E3AD: 85 B9     >867            STA  TXTPTR+1
                >867            <<<
  AF: A0 00     >868  FNCDATA   LDY  #0          ;Retrieve fnc data
E3B1: 68        >869            PLA
E3B2: 91 8A     >871            STA  (FNCNAM),Y
E3B4: 68        >872            PLA
E3B5: C8        >873            INY
E3B6: 91 8A     >871            STA  (FNCNAM),Y
E3B8: 68        >872            PLA
E3B9: C8        >873            INY
E3BA: 91 8A     >871            STA  (FNCNAM),Y
E3BC: 68        >872            PLA
E3BD: C8        >873            INY
E3BE: 91 8A     >871            STA  (FNCNAM),Y
E3C0: 68        >872            PLA
E3C1: C8        >873            INY
E3C2: 91 8A     >875            STA  (FNCNAM),Y
E3C4: 60        >876            RTS
                >877
E3C5: 20 6A DD  >878  STR       JSR  CHKNUM      ;Make sure it is a number
E3C8: A0 00     >879            LDY  #0
E3CA: 20 36 ED  >880            JSR  FACSTRNG    ;Convert to string in stack
E3CD: 68        >881            PLA
E3CE: 68        >882            PLA
E3CF: A9 FF     >883            LDA  #$FF        ;Point to STACK-1
E3D1: A0 00     >884            LDY  #0
E3D3: F0 12     >885            BEQ  STRLIT      ;Create desc & move string
                >886
                >887  * Create string descriptor:
                >888
                >889  STRINI    >>>  TRXY.FAC+3;DSCPTR
E3D5: A6 A0     >889            LDX  FAC+3
E3D7: A4 A1     >889            LDY  FAC+3+1
E3D9: 86 8C     >889            STX  DSCPTR
E3DB: 84 8D     >889            STY  DSCPTR+1
                >889            <<<
E3DD: 20 52 E4  >890  STRSPA    JSR  GETSPA      ;A holds length
E3E0: 86 9E     >891            STX  FAC+1       ;Save descriptor in FAC
E3E2: 84 9F     >892            STY  FAC+2
E3E4: 85 9D     >893            STA  FAC
E3E6: 60        >894            RTS
                >895
E3E7: A2 22     >896  STRLIT    LDX  #'"'
E3E9: 86 0D     >897            STX  CHARAC      ;Set up literal
E3EB: 86 0E     >898            STX  ENDCHR      ; delimiters.
E3ED: 85 AB     >899  STRLT2    STA  STRNG1
E3EF: 84 AC     >900            STY  STRNG1+1
E3F1: 85 9E     >901            STA  FAC+1
E3F3: 84 9F     >902            STY  FAC+2
E3F5: A0 FF     >903            LDY  #$FF
E3F7: C8        >904  FEND      INY              ;Find end of string
E3F8: B1 AB     >905            LDA  (STRNG1),Y
E3FA: F0 0C     >906            BEQ  ZEND
  FC: C5 0D     >907            CMP  CHARAC
E3FE: F0 04     >908            BEQ  QUO?
```

```
E400: C5 0E     >909            CMP  ENDCHR
E402: D0 F3     >910            BNE  FEND
E404: C9 22     >911  QUO?      CMP  #'"'
E406: F0 01     >912            BEQ  NZ
   08: 18       >913  ZEND      CLC
E409: 84 9D     >914  NZ        STY  FAC
E40B: 98        >915            TYA
E40C: 65 AB     >916            ADC  STRNG1
E40E: 85 AD     >917            STA  STRNG2
E410: A6 AC     >918            LDX  STRNG1+1
E412: 90 01     >919            BCC  FE1
E414: E8        >920            INX
E415: 86 AE     >921  FE1       STX  STRNG2+1
E417: A5 AC     >922            LDA  STRNG1+1
E419: F0 04     >923            BEQ  FE2        ;Must move string if it is in
E41B: C9 02     >924            CMP  #2         ; stack or input buffer.
E41D: D0 0B     >925            BNE  PUTNEW
E41F: 98        >926  FE2       TYA
E420: 20 D5 E3  >927            JSR  STRINI
E423: A6 AB     >928            LDX  STRNG1
E425: A4 AC     >929            LDY  STRNG1+1
E427: 20 E2 E5  >930            JSR  MOVSTR
E42A: A6 52     >931  PUTNEW    LDX  TEMPPT
E42C: E0 5E     >932            CPX  #TEMPST+9  ;Too many temp descrs?
E42E: D0 05     >933            BNE  PUTEMP
E430: A2 BF     >934            LDX  #FORMtoCX-ERRMSG
E432: 4C 12 D4  >935  JERR      JMP  ERROR
                >936
E435: A5 9D     >937  PUTEMP    LDA  FAC        ;Set up temp descriptor
   37: 95 00    >938            STA  0,X
   +39: A5 9E   >939            LDA  FAC+1
E43B: 95 01     >940            STA  1,X
E43D: A5 9F     >941            LDA  FAC+2
E43F: 95 02     >942            STA  2,X
E441: A0 00     >943            LDY  #0
E443: 86 A0     >944            STX  FAC+3
E445: 84 A1     >945            STY  FAC+4
E447: 88        >946            DEY
E448: 84 11     >947            STY  VALTYP     ;Flag as string
E44A: 86 53     >948            STX  LASTPT
E44C: E8        >949            INX
E44D: E8        >950            INX
E44E: E8        >951            INX
E44F: 86 52     >952            STX  TEMPPT
E451: 60        >953            RTS
                >954
E452: 46 13     >955  GETSPA    LSR  GARFLG     ;Make space for string
E454: 48        >956  GETSPC    PHA             ;A holds length
E455: 49 FF     >957            EOR  #$FF
E457: 38        >958            SEC
E458: 65 6F     >959            ADC  FRETOP
E45A: A4 70     >960            LDY  FRETOP+1
E45C: B0 01     >961            BCS  CY
E45E: 88        >962            DEY
E45F: C4 6E     >963  CY        CPY  STREND+1
   61: 90 11    >964            BCC  FULL
E463: D0 04     >965            BNE  GOTSPA
```

```
E465: C5 6D     >966             CMP  STREND
E467: 90 0B     >967             BCC  FULL
E469: 85 6F     >968  GOTSPA     STA  FRETOP
E46B: 84 70     >969             STY  FRETOP+1
  6D: 85 71     >970             STA  FRESPC
E46F: 84 72     >971             STY  FRESPC+1
E471: AA        >972             TAX
E472: 68        >973             PLA
E473: 60        >974             RTS
                >975
E474: A2 4D     >976  FULL       LDX  #0ofMEM-ERRMSG
E476: A5 13     >977             LDA  GARFLG      ;Garbage done yet?
E478: 30 B8     >978             BMI  JERR        ;Error if so
E47A: 20 84 E4  >979             JSR  GARBAG
E47D: A9 80     >980             LDA  #$80
E47F: 85 13     >981             STA  GARFLG
E481: 68        >982             PLA
E482: D0 D0     >983             BNE  GETSPC
E484: A6 73     >984  GARBAG     LDX  MEMSIZ      ;Collect from top down
E486: A5 74     >985             LDA  MEMSIZ+1
E488: 86 6F     >986  FNDVAR     STX  FRETOP      ;One pass through all vars
E48A: 85 70     >987             STA  FRETOP+1    ; for each active string!
E48C: A0 00     >988             LDY  #0
E48E: 84 8B     >989             STY  FNCNAM+1
                >990  * Point LOWTR to below collected strings:
                >991             >>>  TRAX.STREND;LOWTR
E490: A5 6D     >991             LDA  STREND
E492: A6 6E     >991             LDX  STREND+1
E494: 85 9B     >991             STA  LOWTR
  96: 86 9C     >991             STX  LOWTR+1
                >991             <<<
E498: A9 55     >992             LDA  #TEMPST
E49A: A2 00     >993             LDX  #>TEMPST
E49C: 85 5E     >994             STA  INDEX
E49E: 86 5F     >995             STX  INDEX+1
E4A0: C5 52     >996  TVAR       CMP  TEMPPT      ;Done with temps?
E4A2: F0 05     >997             BEQ  SVARS       ;Go to simple vars if so
E4A4: 20 23 E5  >998             JSR  DVAR        ;Do a temp
E4A7: F0 F7     >999             BEQ  TVAR        ;Always taken
                >1000
E4A9: A9 07     >1001 SVARS      LDA  #7
E4AB: 85 8F     >1002            STA  DSCLEN
                >1003            >>>  TRAX.VARTAB;INDEX
E4AD: A5 69     >1003            LDA  VARTAB
E4AF: A6 6A     >1003            LDX  VARTAB+1
E4B1: 85 5E     >1003            STA  INDEX
E4B3: 86 5F     >1003            STX  INDEX+1
                >1003            <<<
E4B5: E4 6C     >1004 SVAR       CPX  ARYTAB+1    ;Simple vars done?
E4B7: D0 04     >1005            BNE  SVARGO      ;Continue if not
E4B9: C5 6B     >1006            CMP  ARYTAB
E4BB: F0 05     >1007            BEQ  ARYVAR      ;Do arrays if so
E4BD: 20 19 E5  >1008 SVARGO     JSR  DVARS       ;Do a simple var
E4C0: F0 F3     >1009            BEQ  SVAR        ;Always taken
                >1010
  C2: 85 94     >1011 ARYVAR     STA  ARYPNT
E4C4: 86 95     >1012            STX  ARYPNT+1
```

```
E4C6: A9 03     >1013           LDA  #3
E4C8: 85 8F     >1014           STA  DSCLEN
E4CA: A5 94     >1015 ARYVA2    LDA  ARYPNT
E4CC: A6 95     >1016           LDX  ARYPNT+1
  CE: E4 6E     >1017 ARYVA3    CPX  STREND+1    ;Arrays done?
E4D0: D0 07     >1018           BNE  ARYVGO      ;Do one if not
E4D2: C5 6D     >1019           CMP  STREND
E4D4: D0 03     >1020           BNE  ARYVGO
E4D6: 4C 62 E5  >1021           JMP  GRBPAS      ;All done
                >1022
E4D9: 85 5E     >1023 ARYVGO    STA  INDEX
E4DB: 86 5F     >1024           STX  INDEX+1
E4DD: A0 00     >1025           LDY  #0
E4DF: B1 5E     >1026           LDA  (INDEX),Y
E4E1: AA        >1027           TAX
E4E2: C8        >1028           INY
E4E3: B1 5E     >1029           LDA  (INDEX),Y
E4E5: 08        >1030           PHP
E4E6: C8        >1031           INY
E4E7: B1 5E     >1032           LDA  (INDEX),Y
E4E9: 65 94     >1033           ADC  ARYPNT
E4EB: 85 94     >1034           STA  ARYPNT
E4ED: C8        >1035           INY
E4EE: B1 5E     >1036           LDA  (INDEX),Y
E4F0: 65 95     >1037           ADC  ARYPNT+1
E4F2: 85 95     >1038           STA  ARYPNT+1
E4F4: 28        >1039           PLP
E4F5: 10 D3     >1040           BPL  ARYVA2
E4F7: 8A        >1041           TXA
 4F8: 30 D0     >1042           BMI  ARYVA2
 4FA: C8        >1043           INY
E4FB: B1 5E     >1044           LDA  (INDEX),Y
E4FD: A0 00     >1045           LDY  #0
E4FF: 0A        >1046           ASL
E500: 69 05     >1047           ADC  #5
                >1048           >>>  BUMP.INDEX
E502: 65 5E     >1048           ADC  INDEX
E504: 85 5E     >1048           STA  INDEX
E506: 90 02     >1048           BCC  BPD
E508: E6 5F     >1048           INC  INDEX+1
                >1048 BPD       <<<
E50A: A6 5F     >1049           LDX  INDEX+1
E50C: E4 95     >1050 ARYSTR    CPX  ARYPNT+1    ;Array done?
E50E: D0 04     >1051           BNE  GOGO        ;Do next element if not
E510: C5 94     >1052           CMP  ARYPNT
E512: F0 BA     >1053           BEQ  ARYVA3      ;Next array if so
E514: 20 23 E5  >1054 GOGO      JSR  DVAR
E517: F0 F3     >1055           BEQ  ARYSTR      ;Always taken
                >1056
E519: B1 5E     >1057 DVARS     LDA  (INDEX),Y   ;Integer var?
E51B: 30 35     >1058           BMI  DVARTS      ;Skip if so
E51D: C8        >1059           INY
E51E: B1 5E     >1060           LDA  (INDEX),Y   ;String var?
E520: 10 30     >1061           BPL  DVARTS      ;Skip if not
E522: C8        >1062           INY
  23: B1 5E     >1063 DVAR      LDA  (INDEX),Y   ;Get length
E525: F0 2B     >1064           BEQ  DVARTS      ;Ignore if len 0
```

```
E527: C8        >1065           INY
E528: B1 5E     >1066           LDA  (INDEX),Y
E52A: AA        >1067           TAX
E52B: C8        >1068           INY
  2C: B1 5E     >1069           LDA  (INDEX),Y
E52E: C5 70     >1070           CMP  FRETOP+1
E530: 90 06     >1071           BCC  DV1
E532: D0 1E     >1072           BNE  DVARTS
E534: E4 6F     >1073           CPX  FRETOP
E536: B0 1A     >1074           BCS  DVARTS      ;Skip if collected already
E538: C5 9C     >1075 DV1       CMP  LOWTR+1     ;Above highest string found?
E53A: 90 16     >1076           BCC  DVARTS      ;Skip if not
E53C: D0 04     >1077           BNE  DV2         ;Yes set pointer to it
E53E: E4 9B     >1078           CPX  LOWTR
E540: 90 10     >1079           BCC  DVARTS
E542: 86 9B     >1080 DV2       STX  LOWTR
E544: 85 9C     >1081           STA  LOWTR+1
                >1082           >>>  TRAX.INDEX;FNCNAM
E546: A5 5E     >1082           LDA  INDEX
E548: A6 5F     >1082           LDX  INDEX+1
E54A: 85 8A     >1082           STA  FNCNAM
E54C: 86 8B     >1082           STX  FNCNAM+1
                >1082           <<<
                >1083           >>>  TR.DSCLEN;LENGTH
E54E: A5 8F     >1083           LDA  DSCLEN
E550: 85 91     >1083           STA  LENGTH
                >1083           <<<
E552: A5 8F     >1084 DVARTS    LDA  DSCLEN      ;Set up for next var
E554: 18        >1085           CLC
                >1086           >>>  BUMP.INDEX
  55: 65 5E     >1086           ADC  INDEX
E557: 85 5E     >1086           STA  INDEX
E559: 90 02     >1086           BCC  BPD
E55B: E6 5F     >1086           INC  INDEX+1
                >1086 BPD       <<<
E55D: A6 5F     >1087 VDONE     LDX  INDEX+1
E55F: A0 00     >1088           LDY  #0
E561: 60        >1089           RTS
                >1090
                >1091 * Pass through vars done, now move the highest
                >1092 * string found to top and go back for another.
                >1093
E562: A6 8B     >1094 GRBPAS    LDX  FNCNAM+1    ;Garbage done?
E564: F0 F7     >1095           BEQ  VDONE       ;Yes, return
E566: A5 91     >1096           LDA  LENGTH
E568: 29 04     >1097           AND  #4          ;4 or 2
E56A: 4A        >1098           LSR
E56B: A8        >1099           TAY
E56C: 85 91     >1100           STA  LENGTH      ;2 or 1
                >1101           >>>  AD.(FNCNAM),Y;LOWTR;HIGHTR
E56E: B1 8A     >1101           LDA  (FNCNAM),Y
E570: 65 9B     >1101           ADC  LOWTR
E572: 85 96     >1101           STA  HIGHTR
                >1101           <<<
                >1102           >>>  AD.LOWTR+1;#0;HIGHTR+1
  74: A5 9C     >1102           LDA  LOWTR+1
E576: 69 00     >1102           ADC  #0
```

```
E578: 85 97     >1102          STA  HIGHTR+1
               >1102          <<<
               >1103          >>>  TRAX.FRETOP;ARYPNT
E57A: A5 6F     >1103          LDA  FRETOP
 7C: A6 70     >1103          LDX  FRETOP+1
E57E: 85 94     >1103          STA  ARYPNT
E580: 86 95     >1103          STX  ARYPNT+1
               >1103          <<<
E582: 20 9A D3 >1104          JSR  BLTU2       ;Move string up and
E585: A4 91     >1105          LDY  LENGTH      ; fix its descriptor
E587: C8        >1106          INY
E588: A5 94     >1107          LDA  ARYPNT
E58A: 91 8A     >1108          STA  (FNCNAM),Y
E58C: AA        >1109          TAX
E58D: E6 95     >1110          INC  ARYPNT+1
E58F: A5 95     >1111          LDA  ARYPNT+1
E591: C8        >1112          INY
E592: 91 8A     >1113          STA  (FNCNAM),Y
E594: 4C 88 E4 >1114          JMP  FNDVAR      ;Loop till done
               >1115
               >1116 CAT      >>>  PUSH.FAC+3 ;Save 1st desc ptr ; ENTER WITH TXTPTR
E597: A5 A1     >1116          LDA  FAC+3+1                        on "+" and FACMO,LO
E599: 48        >1116          PHA                                 pointing at first descriptor.
E59A: A5 A0     >1116          LDA  FAC+3
E59C: 48        >1116          PHA
               >1116          <<<
E59D: 20 60 DE >1117          JSR  GETVAL      ; gobble "plus"; evaluate string
E5A0: 20 6C DD >1118          JSR  CHKSTR      ;Get desc ptr to 2nd str
               >1119          >>>  PULL.STRNG1 ;Recover 1st desc ptr
E5A3: 68        >1119          PLA
E5A4: 85 AB     >1119          STA  STRNG1      ;
E5A6: 68        >1119          PLA
E5A7: 85 AC     >1119          STA  STRNG1+1    ; addr of 1st descr.
               >1119          <<<
E5A9: A0 00     >1120          LDY  #0                  get len of 1st string.
E5AB: B1 AB     >1121          LDA  (STRNG1),Y ;Add lengths
E5AD: 18        >1122          CLC             ;↓
E5AE: 71 A0     >1123          ADC  (FAC+3),Y  ; FAC+3 = addr of 2nd descriptor
E5B0: 90 05     >1124          BCC  NTL         ;Ok if < $100
E5B2: A2 B0     >1125          LDX  #STRtoLNG-ERRMSG
E5B4: 4C 12 D4 >1126          JMP  ERROR
               >1127
E5B7: 20 D5 E3 >1128 NTL      JSR  STRINI      ;Get space for concat str
E5BA: 20 D4 E5 >1129          JSR  MOVINS      ;Move 2nd string
E5BD: A5 8C     >1130          LDA  DSCPTR
E5BF: A4 8D     >1131          LDY  DSCPTR+1
E5C1: 20 04 E6 >1132          JSR  FRETMP
E5C4: 20 E6 E5 >1133          JSR  MOVESTR     ;Move 1st string
E5C7: A5 AB     >1134          LDA  STRNG1
E5C9: A4 AC     >1135          LDY  STRNG1+1
E5CB: 20 04 E6 >1136          JSR  FRETMP
E5CE: 20 2A E4 >1137          JSR  PUTNEW      ;Set up desc
E5D1: 4C 95 DD >1138          JMP  FRMEVL2     ;Back for more formula
               >1139
E5D4: A0 00     >1140 MOVINS   LDY  #0
 D6: B1 AB     >1141          LDA  (STRNG1),Y
E5D8: 48        >1142          PHA              ;Length
```

ENTRY: EXPECTS ~~[illegible]~~, acc = combined lengths, ~~(address portion) DESCRIPTOR PTR IN STRNG1 (FRE)~~ PTR to 1st descriptor in STRNG1, FAC holds descriptor for STRING2. When done, new string descr ptd to by FACMO,LO. result flagged as a string

```
E5D9: C8        >1143           INY
E5DA: B1 AB     >1144           LDA  (STRNG1),Y
E5DC: AA        >1145           TAX             ;Put string pointer in X,Y
E5DD: C8        >1146           INY
 DE: B1 AB      >1147           LDA  (STRNG1),Y
E5E0: A8        >1148           TAY
E5E1: 68        >1149           PLA             ;Retrieve length
E5E2: 86 5E     >1150 MOVSTR    STX  INDEX      ;Put pointer in INDEX
E5E4: 84 5F     >1151           STY  INDEX+1
E5E6: A8        >1152 MOVESTR   TAY
E5E7: F0 0A     >1153           BEQ  MVS3
E5E9: 48        >1154           PHA
E5EA: 88        >1155 MVS2      DEY
E5EB: B1 5E     >1156           LDA  (INDEX),Y
E5ED: 91 71     >1157           STA  (FRESPC),Y
E5EF: 98        >1158           TYA
E5F0: D0 F8     >1159           BNE  MVS2
E5F2: 68        >1160           PLA
E5F3: 18        >1161 MVS3      CLC
                >1162           >>>  BUMP.FRESPC
E5F4: 65 71     >1162           ADC  FRESPC
E5F6: 85 71     >1162           STA  FRESPC
E5F8: 90 02     >1162           BCC  BPD
E5FA: E6 72     >1162           INC  FRESPC+1
                >1162 BPD       <<<
E5FC: 60        >1163           RTS
                >1164
E5FD: 20 6C DD  >1165 FRESTR    JSR  CHKSTR     ;Last result a string?
E600: A5 A0     >1166 FREFAC    LDA  FAC+3      ;Get descriptor pointer
 02: A4 A1      >1167           LDY  FAC+4
 04: 85 5E      >1168 FRETMP    STA  INDEX      ;Free temp descriptor
E606: 84 5F     >1169           STY  INDEX+1    ; whose pointer is in (A,Y).
E608: 20 35 E6  >1170           JSR  FRETMS     ;Free descriptor if temp
E60B: 08        >1171           PHP             ;Remember if temp
E60C: A0 00     >1172           LDY  #0
E60E: B1 5E     >1173           LDA  (INDEX),Y
E610: 48        >1174           PHA             ;Push length
E611: C8        >1175           INY
E612: B1 5E     >1176           LDA  (INDEX),Y
E614: AA        >1177           TAX             ;Get pointer to string in X,Y
E615: C8        >1178           INY
E616: B1 5E     >1179           LDA  (INDEX),Y
E618: A8        >1180           TAY
E619: 68        >1181           PLA             ;Retrieve length
E61A: 28        >1182           PLP             ; and status
E61B: D0 13     >1183           BNE  NB         ;Branch if not temp
E61D: C4 70     >1184           CPY  FRETOP+1   ;Was it the lowest string
E61F: D0 0F     >1185           BNE  NB         ; in memory?
E621: E4 6F     >1186           CPX  FRETOP
E623: D0 0B     >1187           BNE  NB         ;Branch if not
E625: 48        >1188           PHA
E626: 18        >1189           CLC             ;"Delete" string if so
                >1190           >>>  BUMP.FRETOP
E627: 65 6F     >1190           ADC  FRETOP
E629: 85 6F     >1190           STA  FRETOP
 2B: 90 02      >1190           BCC  BPD
E62D: E6 70     >1190           INC  FRETOP+1
```

```
                    >1190 BPD       <<<
E62F: 68            >1191           PLA
E630: 86 5E         >1192 NB        STX  INDEX       ;X,Y hold address of string
E632: 84 5F         >1193           STY  INDEX+1     ; and A holds length.
E634: 60            >1194           RTS
                    >1195
E635: C4 54         >1196 FRETMS    CPY  LASTPT+1    ;Free temp descriptor
E637: D0 0C         >1197           BNE  RTN3
E639: C5 53         >1198           CMP  LASTPT
E63B: D0 08         >1199           BNE  RTN3
E63D: 85 52         >1200           STA  TEMPPT
E63F: E9 03         >1201           SBC  #3
E641: 85 53         >1202           STA  LASTPT
E643: A0 00         >1203           LDY  #0
E645: 60            >1204 RTN3      RTS
                    >1205
E646: 20 FB E6      >1206 CHRSTR    JSR  CONINT      ;Convert to byte in X
E649: 8A            >1207           TXA
E64A: 48            >1208           PHA              ;Save it
E64B: A9 01         >1209           LDA  #1          ;Get space for string of length 1
E64D: 20 DD E3      >1210           JSR  STRSPA
E650: 68            >1211           PLA              ;Recall #
E651: A0 00         >1212           LDY  #0          ;Put in string
E653: 91 9E         >1213           STA  (FAC+1),Y
E655: 68            >1214           PLA
E656: 68            >1215           PLA
E657: 4C 2A E4      >1216           JMP  PUTNEW
                    >1217
E65A: 20 B9 E6      >1218 LEFTSTR   JSR  INSTRNG
E65D: D1 8C         >1219           CMP  (DSCPTR),Y
E65F: 98            >1220           TYA
E660: 90 04         >1221 INS1      BCC  INS2
E662: B1 8C         >1222           LDA  (DSCPTR),Y
E664: AA            >1223           TAX
E665: 98            >1224           TYA
E666: 48            >1225 INS2      PHA
E667: 8A            >1226 INS3      TXA
E668: 48            >1227 INS4      PHA
E669: 20 DD E3      >1228           JSR  STRSPA
E66C: A5 8C         >1229           LDA  DSCPTR
E66E: A4 8D         >1230           LDY  DSCPTR+1
E670: 20 04 E6      >1231           JSR  FRETMP
E673: 68            >1232           PLA
E674: A8            >1233           TAY
E675: 68            >1234           PLA
E676: 18            >1235           CLC
                    >1236           >>>  BUMP.INDEX
E677: 65 5E         >1236           ADC  INDEX
E679: 85 5E         >1236           STA  INDEX
E67B: 90 02         >1236           BCC  BPD
E67D: E6 5F         >1236           INC  INDEX+1
                    >1236 BPD       <<<
E67F: 98            >1237           TYA
E680: 20 E6 E5      >1238           JSR  MOVESTR
E683: 4C 2A E4      >1239           JMP  PUTNEW
                    >1240
E686: 20 B9 E6      >1241 RIGHTSTR  JSR  INSTRNG
```

```
E689: 18        >1242           CLC
E68A: F1 8C     >1243           SBC  (DSCPTR),Y
E68C: 49 FF     >1244           EOR  #$FF
E68E: 4C 60 E6  >1245           JMP  INS1
                >1246
E691: A9 FF     >1247 MIDSTR    LDA  #$FF
E693: 85 A1     >1248           STA  FAC+4
E695: 20 B7 00  >1249           JSR  CHRGOT
E698: C9 29     >1250           CMP  #')'
E69A: F0 06     >1251           BEQ  INS5
E69C: 20 BE DE  >1252           JSR  CHKCOM
E69F: 20 F8 E6  >1253           JSR  GETBYT      ;Get 2nd param
E6A2: 20 B9 E6  >1254 INS5      JSR  INSTRNG
E6A5: CA        >1255           DEX
E6A6: 8A        >1256           TXA
E6A7: 48        >1257           PHA
E6A8: 18        >1258           CLC
E6A9: A2 00     >1259           LDX  #0
E6AB: F1 8C     >1260           SBC  (DSCPTR),Y
E6AD: B0 B8     >1261           BCS  INS3
E6AF: 49 FF     >1262           EOR  #$FF
E6B1: C5 A1     >1263           CMP  FAC+4
E6B3: 90 B3     >1264           BCC  INS4
E6B5: A5 A1     >1265           LDA  FAC+4
E6B7: B0 AF     >1266           BCS  INS4        ;Always
                >1267
                >1268 * Common routine for LEFT$, RIGHT$, MID$ to check
                >1269 * for ")", pop return adrs, get descriptor pointer,
                >1270 * and get 1st parameter of command:
                >1271
E6B9: 20 B8 DE  >1272 INSTRNG   JSR  CHKCLS
E6BC: 68        >1273           PLA              ;Pull return address
E6BD: A8        >1274           TAY              ; and save it
E6BE: 68        >1275           PLA
E6BF: 85 91     >1276           STA  LENGTH
E6C1: 68        >1277           PLA              ;Pop previous return adrs
E6C2: 68        >1278           PLA              ; (from GOROUT).
E6C3: 68        >1279           PLA              ;Retrieve 1st parameter
E6C4: AA        >1280           TAX
                >1281           >>>  PULL.DSCPTR
E6C5: 68        >1281           PLA
E6C6: 85 8C     >1281           STA  DSCPTR
E6C8: 68        >1281           PLA
E6C9: 85 8D     >1281           STA  DSCPTR+1
                >1281           <<<
E6CB: A5 91     >1282           LDA  LENGTH
E6CD: 48        >1283           PHA              ;Push back return adrs
E6CE: 98        >1284           TYA
E6CF: 48        >1285           PHA
E6D0: A0 00     >1286           LDY  #0
E6D2: 8A        >1287           TXA              ;Transfer 1st parameter to A
E6D3: F0 1D     >1288           BEQ  GOIQ        ;Error if 0
E6D5: 60        >1289           RTS
                >1290
E6D6: 20 DC E6  >1291 LEN       JSR  GETSTR
E6D9: 4C 01 E3  >1292           JMP  SNGFLT
E6DC: 20 FD E5  >1293 GETSTR    JSR  FRESTR
```

```
E6DF: A2 00     >1294           LDX  #0
E6E1: 86 11     >1295           STX  VALTYP
E6E3: A8        >1296           TAY
E6E4: 60        >1297           RTS
                >1298
E6E5: 20 DC E6  >1299 ASC       JSR  GETSTR      ;Get string ptr in LINNUM
E6E8: F0 08     >1300           BEQ  GOIQ        ;Error if length 0
E6EA: A0 00     >1301           LDY  #0
E6EC: B1 5E     >1302           LDA  (INDEX),Y   ;Get 1st chr of string
E6EE: A8        >1303           TAY
E6EF: 4C 01 E3  >1304           JMP  SNGFLT      ;Float it
E6F2: 4C 99 E1  >1305 GOIQ      JMP  IQERR
                >1306
E6F5: 20 B1 00  >1307 GTBYTC    JSR  CHRGET
E6F8: 20 67 DD  >1308 GETBYT    JSR  FRMNUM
E6FB: 20 08 E1  >1309 CONINT    JSR  MKINT
E6FE: A6 A0     >1310           LDX  FAC+3       ;<256?
E700: D0 F0     >1311           BNE  GOIQ        ;Error if not
E702: A6 A1     >1312           LDX  FAC+4
E704: 4C B7 00  >1313           JMP  CHRGOT
                >1314
E707: 20 DC E6  >1315 VAL       JSR  GETSTR      ;Get pointer to string in INDEX
E70A: D0 03     >1316           BNE  VL2
E70C: 4C 4E E8  >1317           JMP  ZEROFAC     ;Return 0 if length=0
                >1318 VL2       >>>  TRXY.TXTPTR;STRNG2
E70F: A6 B8     >1318           LDX  TXTPTR
E711: A4 B9     >1318           LDY  TXTPTR+1
E713: 86 AD     >1318           STX  STRNG2
E715: 84 AE     >1318           STY  STRNG2+1
                >1318           <<<
E717: A6 5E     >1319           LDX  INDEX
E719: 86 B8     >1320           STX  TXTPTR      ;Point TXTPTR to start of string
E71B: 18        >1321           CLC
E71C: 65 5E     >1322           ADC  INDEX       ;Add length
E71E: 85 60     >1323           STA  DEST        ;Point DEST to end of stg + 1
E720: A6 5F     >1324           LDX  INDEX+1
E722: 86 B9     >1325           STX  TXTPTR+1
E724: 90 01     >1326           BCC  VL3
E726: E8        >1327           INX
E727: 86 61     >1328 VL3       STX  DEST+1
E729: A0 00     >1329           LDY  #0
E72B: B1 60     >1330           LDA  (DEST),Y    ;Get byte following string
E72D: 48        >1331           PHA              ;Save it
E72E: A9 00     >1332           LDA  #0
E730: 91 60     >1333           STA  (DEST),Y    ;Put 0 there
E732: 20 B7 00  >1334           JSR  CHRGOT
E735: 20 4A EC  >1335           JSR  FIN         ;Evaluate string
E738: 68        >1336           PLA
E739: A0 00     >1337           LDY  #0
E73B: 91 60     >1338           STA  (DEST),Y    ;Replace byte at end
                >1339 POINT     >>>  TRXY.STRNG2;TXTPTR
E73D: A6 AD     >1339           LDX  STRNG2
E73F: A4 AE     >1339           LDY  STRNG2+1
E741: 86 B8     >1339           STX  TXTPTR
E743: 84 B9     >1339           STY  TXTPTR+1
                >1339           <<<
E745: 60        >1340           RTS
```

III

```
                     >1341
E746: 20 67 DD       >1342 GTNUM     JSR  FRMNUM
E749: 20 52 E7       >1343           JSR  GETADR
E74C: 20 BE DE       >1344 COMBYTE   JSR  CHKCOM
  4F: 4C F8 E6       >1345           JMP  GETBYT
                     >1346
E752: A5 9D          >1347 GETADR    LDA  FAC          ;FAC <= $FFFF?
E754: C9 91          >1348           CMP  #$91
E756: B0 9A          >1349           BGE  GOIQ         ;Error if not.
E758: 20 F2 EB       >1350           JSR  QINT         ;Convert to integer
E75B: A5 A0          >1351           LDA  FAC+3        ; and move it
E75D: A4 A1          >1352           LDY  FAC+4
E75F: 84 50          >1353           STY  LINNUM       ; to LINNUM
E761: 85 51          >1354           STA  LINNUM+1
E763: 60             >1355           RTS
                     >1356
E764: A5 50          >1357 PEEK      LDA  LINNUM       ;Protect LINNUM
E766: 48             >1358           PHA
E767: A5 51          >1359           LDA  LINNUM+1
E769: 48             >1360           PHA
E76A: 20 52 E7       >1361           JSR  GETADR
E76D: A0 00          >1362           LDY  #0
E76F: B1 50          >1363           LDA  (LINNUM),Y
E771: A8             >1364           TAY
E772: 68             >1365           PLA               ;Retrieve LINNUM
E773: 85 51          >1366           STA  LINNUM+1
E775: 68             >1367           PLA
E776: 85 50          >1368           STA  LINNUM
E778: 4C 01 E3       >1369           JMP  SNGFLT
                     >1370
  7B: 20 46 E7       >1371 POKE      JSR  GTNUM
E77E: 8A             >1372           TXA
E77F: A0 00          >1373           LDY  #0
E781: 91 50          >1374           STA  (LINNUM),Y
E783: 60             >1375           RTS
                     >1376
E784: 20 46 E7       >1377 WAIT      JSR  GTNUM        ;Get address in LINNUM
E787: 86 85          >1378           STX  FORPNT       ; & specified mask in FORPNT
E789: A2 00          >1379           LDX  #0
E78B: 20 B7 00       >1380           JSR  CHRGOT       ;Inversion byte specified?
E78E: F0 03          >1381           BEQ  WT2          ;Branch if not
E790: 20 4C E7       >1382           JSR  COMBYTE      ;Get it
E793: 86 86          >1383 WT2       STX  FORPNT+1     ;Set up inversion byte
E795: A0 00          >1384           LDY  #0
E797: B1 50          >1385 WT3       LDA  (LINNUM),Y   ;Get byte at address
E799: 45 86          >1386           EOR  FORPNT+1     ;Invert as specified
E79B: 25 85          >1387           AND  FORPNT       ;Mask it
E79D: F0 F8          >1388           BEQ  WT3          ;Loop till not 0
E79F: 60             >1389 RTN4      RTS
                     >1390
                      152            PUT  APSOFT III
                     >1     ******************************
                     >2     *                            *
                     >3     *   Applesoft - Part III     *
                     >4     *                            *
                     >5     * Copywrite Apple Computer, *
                     >6     * Inc. and Microsoft, Inc.; *
```

```
                >7      * not for publication or     *
                >8      * distribution.              *
                >9      *                            *
                >10     ******************************
                >11     *                            *
                >12     *  Floating Point Routines   *
                >13     *                            *
                >14     *       $E7A0 - $F1D4        *
                >15     *                            *
                >16     ******************************
                >17
E7A0: A9 64     >18     FADDH   LDA  #HALF      ;FAC+1/2 -> FAC
E7A2: A0 EE     >19             LDY  #>HALF
E7A4: 4C BE E7  >20             JMP  FADD
                >21
E7A7: 20 E3 E9  >22     FSUB    JSR  CONUPK     ;Load ARG from (A,Y)
                >23     FSUBT   >>>  NEG.FACSGN ;ARG - FAC -> FAC
E7AA: A5 A2     >23             LDA  FACSGN
E7AC: 49 FF     >23             EOR  #$FF
E7AE: 85 A2     >23             STA  FACSGN
                >23             <<<
E7B0: 45 AA     >24             EOR  ARGSGN
E7B2: 85 AB     >25             STA  SGNCPR
E7B4: A5 9D     >26             LDA  FAC
E7B6: 4C C1 E7  >27             JMP  FADDT
                >28
E7B9: 20 F0 E8  >29     ADO     JSR  SHIFT
E7BC: 90 3C     >30             BCC  AD5
E7BE: 20 E3 E9  >31     FADD    JSR  CONUPK     ;(A,Y) to ARG
E7C1: D0 03     >32     FADDT   BNE  AD1        ;ARG + FAC -> FAC
E7C3: 4C 53 EB  >33             JMP  MOVFA      ;If FAC=0 just move ARG over
E7C6: A6 AC     >34     AD1     LDX  EXTRAFAC   ;Extra byte for precision
E7C8: 86 92     >35             STX  EXTRASV    ; in all FP routines.
E7CA: A2 A5     >36             LDX  #ARG       ;Set up to shift ARG
E7CC: A5 A5     >37             LDA  ARG
E7CE: A8        >38     AD2     TAY
E7CF: F0 CE     >39             BEQ  RTN4       ;If ARG=0 exit
E7D1: 38        >40             SEC
E7D2: E5 9D     >41             SBC  FAC        ;Get diffnce of exp
E7D4: F0 24     >42             BEQ  AD5        ;Go add if same exp
E7D6: 90 12     >43             BCC  AD3
E7D8: 84 9D     >44             STY  FAC        ;Sneaky exchange
E7DA: A4 AA     >45             LDY  ARGSGN
E7DC: 84 A2     >46             STY  FACSGN
E7DE: 49 FF     >47             EOR  #$FF
E7E0: 69 00     >48             ADC  #0
E7E2: A0 00     >49             LDY  #0
E7E4: 84 92     >50             STY  EXTRASV
E7E6: A2 9D     >51             LDX  #FAC       ;Set up to shift FAC
E7E8: D0 04     >52             BNE  AD4
E7EA: A0 00     >53     AD3     LDY  #0
E7EC: 84 AC     >54             STY  EXTRAFAC
E7EE: C9 F9     >55     AD4     CMP  #$F9       ;How many bits to shift?
E7F0: 30 C7     >56             BMI  ADO        ;Branch if more than 7
E7F2: A8        >57             TAY             ;Index to # of shifts
E7F3: A5 AC     >58             LDA  EXTRAFAC
E7F5: 56 01     >59             LSR  1,X
```

```
E7F7: 20 07 E9 >60             JSR  SHFTR       ;Do shift
E7FA: 24 AB    >61   AD5       BIT  SGNCPR      ;Same sign?
E7FC: 10 57    >62             BPL  ADMAN       ;Yes, add mantissas
E7FE: A0 9D    >63             LDY  #FAC
  00: E0 A5    >64             CPX  #ARG        ;Which was adjusted?
E802: F0 02    >65             BEQ  SUBMAN      ;If ARG, do FAC-ARG
E804: A0 A5    >66             LDY  #ARG        ;If FAC, do ARG-FAC
E806: 38       >67   SUBMAN    SEC
E807: 49 FF    >68             EOR  #$FF
E809: 65 92    >69             ADC  EXTRASV
E80B: 85 AC    >70             STA  EXTRAFAC
               >71             >>>  SB.4,Y;4,X;FAC+4
E80D: B9 04 00 >71             LDA  4,Y
E810: F5 04    >71             SBC  4,X
E812: 85 A1    >71             STA  FAC+4
               >71             <<<
               >72             >>>  SB.3,Y;3,X;FAC+3
E814: B9 03 00 >72             LDA  3,Y
E817: F5 03    >72             SBC  3,X
E819: 85 A0    >72             STA  FAC+3
               >72             <<<
               >73             >>>  SB.2,Y;2,X;FAC+2
E81B: B9 02 00 >73             LDA  2,Y
E81E: F5 02    >73             SBC  2,X
E820: 85 9F    >73             STA  FAC+2
               >73             <<<
               >74             >>>  SB.1,Y;1,X;FAC+1
E822: B9 01 00 >74             LDA  1,Y
E825: F5 01    >74             SBC  1,X
927: 85 9E     >74             STA  FAC+1
               >74             <<<
E829: B0 03    >75   SGNIF     BCS  SIGNIF
E82B: 20 9E E8 >76             JSR  NEGFAC
E82E: A0 00    >77   SIGNIF    LDY  #0          ;Shift up signif digit
E830: 98       >78             TYA              ;Counting shift in A
E831: 18       >79             CLC
E832: A6 9E    >80   FLOOP     LDX  FAC+1
E834: D0 4A    >81             BNE  FR2         ;Repeat till not 0
E836: A6 9F    >82             LDX  FAC+2
E838: 86 9E    >83             STX  FAC+1
E83A: A6 A0    >84             LDX  FAC+3
E83C: 86 9F    >85             STX  FAC+2
E83E: A6 A1    >86             LDX  FAC+4
E840: 86 A0    >87             STX  FAC+3
E842: A6 AC    >88             LDX  EXTRAFAC
E844: 86 A1    >89             STX  FAC+4
E846: 84 AC    >90             STY  EXTRAFAC    ;Zero extra byte
E848: 69 08    >91             ADC  #8
E84A: C9 20    >92             CMP  #8*4        ;Done 4 times?
E84C: D0 E4    >93             BNE  FLOOP       ;Loop if not
E84E: A9 00    >94   ZEROFAC   LDA  #0
E850: 85 9D    >95   AtoFAC    STA  FAC
E852: 85 A2    >96   AtoFACS   STA  FACSGN
E854: 60       >97             RTS
               >98
055: 65 92     >99   ADMAN     ADC  EXTRASV     ;Add mantissas (frac parts)
E857: 85 AC    >100            STA  EXTRAFAC
```

```
                        >101            >>>  AD.FAC+4;ARG+4;FAC+4
E859: A5 A1             >101            LDA  FAC+4
E85B: 65 A9             >101            ADC  ARG+4
E85D: 85 A1             >101            STA  FAC+4
                        >101            <<<
                        >102            >>>  AD.FAC+3;ARG+3;FAC+3
E85F: A5 A0             >102            LDA  FAC+3
E861: 65 A8             >102            ADC  ARG+3
E863: 85 A0             >102            STA  FAC+3
                        >102            <<<
                        >103            >>>  AD.FAC+2;ARG+2;FAC+2
E865: A5 9F             >103            LDA  FAC+2
E867: 65 A7             >103            ADC  ARG+2
E869: 85 9F             >103            STA  FAC+2
                        >103            <<<
                        >104            >>>  AD.FAC+1;ARG+1;FAC+1
E86B: A5 9E             >104            LDA  FAC+1
E86D: 65 A6             >104            ADC  ARG+1
E86F: 85 9E             >104            STA  FAC+1
                        >104            <<<
E871: 4C 8D E8          >105            JMP  FR3
                        >106
E874: 69 01             >107  FR1       ADC  #1          ;Count bits shifted
E876: 06 AC             >108            ASL  EXTRAFAC
E878: 26 A1             >109            ROL  FAC+4
E87A: 26 A0             >110            ROL  FAC+3
E87C: 26 9F             >111            ROL  FAC+2
E87E: 26 9E             >112            ROL  FAC+1
E880: 10 F2             >113  FR2       BPL  FR1         ;Repeat till FAC+1 neg
E882: 38                >114            SEC
E883: E5 9D             >115            SBC  FAC         ;Fix exponent
E885: B0 C7             >116            BCS  ZEROFAC
E887: 49 FF             >117            EOR  #$FF
E889: 69 01             >118            ADC  #1
E88B: 85 9D             >119            STA  FAC
E88D: 90 0E             >120  FR3       BCC  RTN5
E88F: E6 9D             >121  FROUND    INC  FAC
E891: F0 42             >122            BEQ  OVERFLOW
E893: 66 9E             >123            ROR  FAC+1
E895: 66 9F             >124            ROR  FAC+2
E897: 66 A0             >125            ROR  FAC+3
E899: 66 A1             >126            ROR  FAC+4
E89B: 66 AC             >127            ROR  EXTRAFAC
E89D: 60                >128  RTN5      RTS
                        >129
                        >130  NEGFAC    >>>  NEG.FACSGN
E89E: A5 A2             >130            LDA  FACSGN
E8A0: 49 FF             >130            EOR  #$FF
E8A2: 85 A2             >130            STA  FACSGN
                        >130            <<<
                        >131  NEG2      >>>  NEG.FAC+1
E8A4: A5 9E             >131            LDA  FAC+1
E8A6: 49 FF             >131            EOR  #$FF
E8A8: 85 9E             >131            STA  FAC+1
                        >131            <<<
                        >132            >>>  NEG.FAC+2
E8AA: A5 9F             >132            LDA  FAC+2
```

```
E8AC: 49 FF     >132            EOR  #$FF
E8AE: 85 9F     >132            STA  FAC+2
                >132            <<<
                >133            >>>  NEG.FAC+3
  B0: A5 A0     >133            LDA  FAC+3
E8B2: 49 FF     >133            EOR  #$FF
E8B4: 85 A0     >133            STA  FAC+3
                >133            <<<
                >134            >>>  NEG.FAC+4
E8B6: A5 A1     >134            LDA  FAC+4
E8B8: 49 FF     >134            EOR  #$FF
E8BA: 85 A1     >134            STA  FAC+4
                >134            <<<
                >135            >>>  NEG.EXTRAFAC
E8BC: A5 AC     >135            LDA  EXTRAFAC
E8BE: 49 FF     >135            EOR  #$FF
E8C0: 85 AC     >135            STA  EXTRAFAC
                >135            <<<
E8C2: E6 AC     >136            INC  EXTRAFAC
E8C4: D0 0E     >137            BNE  RTN6
E8C6: E6 A1     >138  PLUSEPS   INC  FAC+4      ;Add carry from EXTRA
E8C8: D0 0A     >139            BNE  RTN6
E8CA: E6 A0     >140            INC  FAC+3
E8CC: D0 06     >141            BNE  RTN6
E8CE: E6 9F     >142            INC  FAC+2
E8D0: D0 02     >143            BNE  RTN6
E8D2: E6 9E     >144            INC  FAC+1
E8D4: 60        >145  RTN6      RTS
                >146
  D5: A2 45     >147  OVERFLOW  LDX  #OVFLOW-ERRMSG
  D7: 4C 12 D4  >148            JMP  ERROR
                >149
E8DA: A2 61     >150  SHFTRES   LDX  #RESULT-1
E8DC: B4 04     >151  NXSFT     LDY  4,X
E8DE: 84 AC     >152            STY  EXTRAFAC
E8E0: B4 03     >153            LDY  3,X
E8E2: 94 04     >154            STY  4,X
E8E4: B4 02     >155            LDY  2,X
E8E6: 94 03     >156            STY  3,X
E8E8: B4 01     >157            LDY  1,X
E8EA: 94 02     >158            STY  2,X
E8EC: A4 A4     >159            LDY  FPGEN
E8EE: 94 01     >160            STY  1,X
E8F0: 69 08     >161  SHIFT     ADC  #8         ;Shift 1,X right #bits in A
E8F2: 30 E8     >162            BMI  NXSFT
E8F4: F0 E6     >163            BEQ  NXSFT
E8F6: E9 08     >164            SBC  #8
E8F8: A8        >165            TAY
E8F9: A5 AC     >166            LDA  EXTRAFAC
E8FB: B0 14     >167            BCS  SH3
E8FD: 16 01     >168  SH1       ASL  1,X
E8FF: 90 02     >169            BCC  SH2
E901: F6 01     >170            INC  1,X
E903: 76 01     >171  SH2       ROR  1,X
E905: 76 01     >172            ROR  1,X
  07: 76 02     >173  SHFTR     ROR  2,X
E909: 76 03     >174            ROR  3,X
```

```
E90B: 76 04     >175            ROR  4,X
E90D: 6A        >176            ROR
E90E: C8        >177            INY
E90F: D0 EC     >178            BNE  SH1
  11: 18        >179  SH3       CLC
E912: 60        >180            RTS
                >181
E913: 81 00 00
E916: 00 00     >182  ONE       HEX  8100000000
E918: 03        >183  LOGSER    DFB  3           ;Index to # of coefs:
E919: 7F 5E 56
E91C: CB 79     >184            HEX  7F5E56CB79
E91E: 80 13 9B
E921: 0B 64     >185            HEX  80139B0B64
E923: 80 76 38
E926: 93 16     >186            HEX  8076389316
E928: 82 38 AA
E92B: 3B 20     >187            HEX  8238AA3B20
E92D: 80 35 04
E930: F3 34     >188  SQRhalf   HEX  803504F334
E932: 81 35 04
E935: F3 34     >189  SQRtwo    HEX  813504F334
E937: 80 80 00
E93A: 00 00     >190  HALFneg   HEX  8080000000
E93C: 80 31 72
E93F: 17 F8     >191  LOGtwo    HEX  80317217F8
                >192
E941: 20 82 EB >193  LOG        JSR  SIGN        ;Natural log of FAC
E944: F0 02    >194             BEQ  GIQ         ;Argument must be +
 46: 10 03     >195             BPL  LG2
 48: 4C 99 E1  >196  GIQ        JMP  IQERR
E94B: A5 9D    >197  LG2        LDA  FAC
E94D: E9 7F    >198             SBC  #$7F        ;Save exponent
E94F: 48       >199             PHA
E950: A9 80    >200             LDA  #$80        ;Normalize between .5 and 1
E952: 85 9D    >201             STA  FAC
E954: A9 2D    >202             LDA  #SQRhalf
E956: A0 E9    >203             LDY  #>SQRhalf
E958: 20 BE E7 >204             JSR  FADD        ;Compute via series of odd
E95B: A9 32    >205             LDA  #SQRtwo     ; powers of
E95D: A0 E9    >206             LDY  #>SQRtwo    ; (SQR(2)X-1)/(SQR(2)X+1)
E95F: 20 66 EA >207             JSR  FDIV
E962: A9 13    >208             LDA  #ONE
E964: A0 E9    >209             LDY  #>ONE
E966: 20 A7 E7 >210             JSR  FSUB
E969: A9 18    >211             LDA  #LOGSER
E96B: A0 E9    >212             LDY  #>LOGSER
E96D: 20 5C EF >213             JSR  ODDSER
E970: A9 37    >214             LDA  #HALFneg
E972: A0 E9    >215             LDY  #>HALFneg
E974: 20 BE E7 >216             JSR  FADD
E977: 68       >217             PLA
E978: 20 D5 EC >218             JSR  ADDACC      ;Add original exponent
E97B: A9 3C    >219             LDA  #LOGtwo
E97D: A0 E9    >220             LDY  #>LOGtwo
 7F: 20 E3 E9 >221  FMULT       JSR  CONUPK      ;(A,Y) to ARG
E982: D0 03    >222  FMULTT     BNE  FMU         ;FAC * ARG -> FAC
```

```
E984: 4C E2 E9 >223              JMP  RTN7
              >224
E987: 20 0E EA >225  FMU         JSR  ADEXP
E98A: A9 00    >226              LDA  #0
E98C: 85 62    >227              STA  RESULT
E98E: 85 63    >228              STA  RESULT+1
E990: 85 64    >229              STA  RESULT+2
E992: 85 65    >230              STA  RESULT+3
E994: A5 AC    >231              LDA  EXTRAFAC
E996: 20 B0 E9 >232              JSR  FM1
E999: A5 A1    >233              LDA  FAC+4
E99B: 20 B0 E9 >234              JSR  FM1
E99E: A5 A0    >235              LDA  FAC+3
E9A0: 20 B0 E9 >236              JSR  FM1
E9A3: A5 9F    >237              LDA  FAC+2
E9A5: 20 B0 E9 >238              JSR  FM1
E9A8: A5 9E    >239              LDA  FAC+1
E9AA: 20 B5 E9 >240              JSR  FM2
E9AD: 4C E6 EA >241              JMP  RES>FAC
              >242
E9B0: D0 03    >243  FM1         BNE  FM2
E9B2: 4C DA E8 >244              JMP  SHFTRES
              >245
E9B5: 4A       >246  FM2         LSR
E9B6: 09 80    >247              ORA  #$80
E9B8: A8       >248  FM3         TAY
E9B9: 90 19    >249              BCC  FM4
E9BB: 18       >250              CLC
              >251              >>>  AD.RESULT+3;ARG+4;RESULT+3
E9BC: A5 65    >251              LDA  RESULT+3
E9BE: 65 A9    >251              ADC  ARG+4
E9C0: 85 65    >251              STA  RESULT+3
              >251              <<<
              >252              >>>  AD.RESULT+2;ARG+3;RESULT+2
E9C2: A5 64    >252              LDA  RESULT+2
E9C4: 65 A8    >252              ADC  ARG+3
E9C6: 85 64    >252              STA  RESULT+2
              >252              <<<
              >253              >>>  AD.RESULT+1;ARG+2;RESULT+1
E9C8: A5 63    >253              LDA  RESULT+1
E9CA: 65 A7    >253              ADC  ARG+2
E9CC: 85 63    >253              STA  RESULT+1
              >253              <<<
              >254              >>>  AD.RESULT;ARG+1;RESULT
E9CE: A5 62    >254              LDA  RESULT
E9D0: 65 A6    >254              ADC  ARG+1
E9D2: 85 62    >254              STA  RESULT
              >254              <<<
E9D4: 66 62    >255  FM4         ROR  RESULT
E9D6: 66 63    >256              ROR  RESULT+1
E9D8: 66 64    >257              ROR  RESULT+2
E9DA: 66 65    >258              ROR  RESULT+3
E9DC: 66 AC    >259              ROR  EXTRAFAC
E9DE: 98       >260              TYA
E9DF: 4A       >261              LSR
E9E0: D0 D6    >262              BNE  FM3
E9E2: 60       >263  RTN7        RTS
```

```
                        >264
                        >265  * Unpack number at (A,Y) and move to ARG:
                        >266
E9E3: 85 5E             >267  CONUPK    STA  INDEX
  E5: 84 5F             >268           STY  INDEX+1
E9E7: A0 04             >269           LDY  #4
E9E9: B1 5E             >270           LDA  (INDEX),Y
E9EB: 85 A9             >271           STA  ARG+4
E9ED: 88                >272           DEY
E9EE: B1 5E             >273           LDA  (INDEX),Y
E9F0: 85 A8             >274           STA  ARG+3
E9F2: 88                >275           DEY
E9F3: B1 5E             >276           LDA  (INDEX),Y
E9F5: 85 A7             >277           STA  ARG+2
E9F7: 88                >278           DEY
E9F8: B1 5E             >279           LDA  (INDEX),Y
E9FA: 85 AA             >280           STA  ARGSGN
E9FC: 45 A2             >281           EOR  FACSGN
E9FE: 85 AB             >282           STA  SGNCPR
EA00: A5 AA             >283           LDA  ARGSGN
EA02: 09 80             >284           ORA  #$80
EA04: 85 A6             >285           STA  ARG+1
EA06: 88                >286           DEY
EA07: B1 5E             >287           LDA  (INDEX),Y
EA09: 85 A5             >288           STA  ARG
EA0B: A5 9D             >289           LDA  FAC
EA0D: 60                >290           RTS
                        >291
EA0E: A5 A5             >292  ADEXP     LDA  ARG
 10: F0 1F              >293  ADEX2     BEQ  ZERO
 12: 18                 >294           CLC
EA13: 65 9D             >295           ADC  FAC
EA15: 90 04             >296           BCC  ADEX3
EA17: 30 1D             >297           BMI  JOV
EA19: 18                >298           CLC
EA1A: 2C                >299           HEX  2C          ;Trick to branch
EA1B: 10 14             >300  ADEX3     BPL  ZERO
EA1D: 69 80             >301           ADC  #$80
EA1F: 85 9D             >302           STA  FAC
EA21: D0 03             >303           BNE  ADEX4
EA23: 4C 52 E8          >304           JMP  AtoFACS
EA26: A5 AB             >305  ADEX4     LDA  SGNCPR
EA28: 85 A2             >306           STA  FACSGN
EA2A: 60                >307           RTS
                        >308
EA2B: A5 A2             >309  OUTOFRNG LDA  FACSGN
EA2D: 49 FF             >310           EOR  #$FF
EA2F: 30 05             >311           BMI  JOV         ;Error if positive #
EA31: 68                >312  ZERO      PLA
EA32: 68                >313           PLA
EA33: 4C 4E E8          >314           JMP  ZEROFAC     ;Return 0 if negative #
EA36: 4C D5 E8          >315  JOV       JMP  OVERFLOW
                        >316
                        >317  * Routine to multiply FAC by 10:
                        >318
 39: 20 63 EB           >319  MUL10     JSR  MOVAF       ;Copy FAC to ARG
EA3C: AA                >320           TAX              ;A-reg holds FAC
```

```
EA3D: F0 10     >321            BEQ  RTN8       ;Exit if FAC=0
EA3F: 18        >322            CLC
EA40: 69 02     >323            ADC  #2         ;Simulate *4
EA42: B0 F2     >324            BCS  JOV
EA44: A2 00     >325            LDX  #0
EA46: 86 AB     >326            STX  SGNCPR
EA48: 20 CE E7  >327            JSR  AD2        ;FAC*4 + ARG -> FAC
EA4B: E6 9D     >328            INC  FAC        ;= mult by 2
EA4D: F0 E7     >329            BEQ  JOV
EA4F: 60        >330  RTN8      RTS
                >331
EA50: 84 20 00
EA53: 00 00     >332  NUM10     HEX  8420000000
                >333
                >334  * Routine to divide FAC by 10:
                >335
EA55: 20 63 EB  >336  DIV10     JSR  MOVAF      ;Copy FAC to ARG
EA58: A9 50     >337            LDA  #NUM10     ;Set up to put
EA5A: A0 EA     >338            LDY  #>NUM10    ; 10 in FAC
EA5C: A2 00     >339            LDX  #0
EA5E: 86 AB     >340  DIV       STX  SGNCPR
EA60: 20 F9 EA  >341            JSR  MOVFM      ;Put (A,Y) in FAC
EA63: 4C 69 EA  >342            JMP  FDIVT      ;Divide ARG by FAC
                >343
EA66: 20 E3 E9  >344  FDIV      JSR  CONUPK     ;(A,Y) -> ARG
EA69: F0 76     >345  FDIVT     BEQ  DIVZ       ;ARG/FAC -> FAC
EA6B: 20 72 EB  >346            JSR  RNDB
EA6E: A9 00     >347            LDA  #0
EA70: 38        >348            SEC
EA71: E5 9D     >349            SBC  FAC
EA73: 85 9D     >350            STA  FAC
EA75: 20 0E EA  >351            JSR  ADEXP
EA78: E6 9D     >352            INC  FAC
EA7A: F0 BA     >353            BEQ  JOV
EA7C: A2 FC     >354            LDX  #$100-4    ;Looping index
EA7E: A9 01     >355            LDA  #1
EA80: A4 A6     >356  FD1       LDY  ARG+1
EA82: C4 9E     >357            CPY  FAC+1
EA84: D0 10     >358            BNE  FD2
EA86: A4 A7     >359            LDY  ARG+2
EA88: C4 9F     >360            CPY  FAC+2
EA8A: D0 0A     >361            BNE  FD2
EA8C: A4 A8     >362            LDY  ARG+3
EA8E: C4 A0     >363            CPY  FAC+3
EA90: D0 04     >364            BNE  FD2
EA92: A4 A9     >365            LDY  ARG+4
EA94: C4 A1     >366            CPY  FAC+4
EA96: 08        >367  FD2       PHP
EA97: 2A        >368            ROL
EA98: 90 09     >369            BCC  FD3
EA9A: E8        >370            INX
EA9B: 95 65     >371            STA  RESULT+3,X
EA9D: F0 32     >372            BEQ  FD6
EA9F: 10 34     >373            BPL  FD7        ;Final exit when X=1
EAA1: A9 01     >374            LDA  #1
EAA3: 28        >375  FD3       PLP
EAA4: B0 0E     >376            BCS  FD5
```

```
EAA6: 06 A9     >377  FD4       ASL  ARG+4
EAA8: 26 A8     >378            ROL  ARG+3
EAAA: 26 A7     >379            ROL  ARG+2
EAAC: 26 A6     >380            ROL  ARG+1
  AE: B0 E6     >381            BCS  FD2
EAB0: 30 CE     >382            BMI  FD1
EAB2: 10 E2     >383            BPL  FD2
EAB4: A8        >384  FD5       TAY
                >385            >>>  SB.ARG+4;FAC+4;ARG+4
EAB5: A5 A9     >385            LDA  ARG+4
EAB7: E5 A1     >385            SBC  FAC+4
EAB9: 85 A9     >385            STA  ARG+4
                >385            <<<
                >386            >>>  SB.ARG+3;FAC+3;ARG+3
EABB: A5 A8     >386            LDA  ARG+3
EABD: E5 A0     >386            SBC  FAC+3
EABF: 85 A8     >386            STA  ARG+3
                >386            <<<
                >387            >>>  SB.ARG+2;FAC+2;ARG+2
EAC1: A5 A7     >387            LDA  ARG+2
EAC3: E5 9F     >387            SBC  FAC+2
EAC5: 85 A7     >387            STA  ARG+2
                >387            <<<
                >388            >>>  SB.ARG+1;FAC+1;ARG+1
EAC7: A5 A6     >388            LDA  ARG+1
EAC9: E5 9E     >388            SBC  FAC+1
EACB: 85 A6     >388            STA  ARG+1
                >388            <<<
EACD: 98        >389            TYA
CE: 4C A6 EA    >390            JMP  FD4
D1: A9 40       >391  FD6       LDA  #$40
EAD3: D0 CE     >392            BNE  FD3
                >393  FD7       LUP  6
EAD5: 0A        >394            ASL
EAD6: 0A        >394            ASL
EAD7: 0A        >394            ASL
EAD8: 0A        >394            ASL
EAD9: 0A        >394            ASL
EADA: 0A        >394            ASL
EADB: 85 AC     >396            STA  EXTRAFAC
EADD: 28        >397            PLP
EADE: 4C E6 EA  >398            JMP  RES>FAC
EAE1: A2 85     >399  DIVZ      LDX  #DIVbyZRO-ERRMSG
EAE3: 4C 12 D4  >400            JMP  ERROR
                >401  RES>FAC   >>>  TRDB.RESULT;FAC+1
                >401            >>>  TR.RESULT;FAC+1
EAE6: A5 62     >401            LDA  RESULT
EAE8: 85 9E     >401            STA  FAC+1
                >401            <<<
                >401            >>>  TR.RESULT+1;FAC+1+1
EAEA: A5 63     >401            LDA  RESULT+1
EAEC: 85 9F     >401            STA  FAC+1+1
                >401            <<<
                >401            <<<
                >402            >>>  TRDB.RESULT+2;FAC+3
                >402            >>>  TR.RESULT+2;FAC+3
EAEE: A5 64     >402            LDA  RESULT+2
```

```
EAF0: 85 A0     >402            STA  FAC+3
                >402            <<<
                >402            >>>  TR.RESULT+2+1;FAC+3+1
EAF2: A5 65     >402            LDA  RESULT+2+1
EAF4: 85 A1     >402            STA  FAC+3+1
                >402            <<<
                >402            <<<
EAF6: 4C 2E E8  >403            JMP  SIGNIF
                >404
                >405  * Routine to get packed floating # at (A,Y)
                >406  * unpack it and move it to FAC:
                >407
EAF9: 85 5E     >408  MOVFM     STA  INDEX
EAFB: 84 5F     >409            STY  INDEX+1
EAFD: A0 04     >410            LDY  #4
EAFF: B1 5E     >411            LDA  (INDEX),Y
EB01: 85 A1     >412            STA  FAC+4
EB03: 88        >413            DEY
EB04: B1 5E     >414            LDA  (INDEX),Y
EB06: 85 A0     >415            STA  FAC+3
EB08: 88        >416            DEY
EB09: B1 5E     >417            LDA  (INDEX),Y
EB0B: 85 9F     >418            STA  FAC+2
EB0D: 88        >419            DEY
EB0E: B1 5E     >420            LDA  (INDEX),Y
EB10: 85 A2     >421            STA  FACSGN      ;Unpack
EB12: 09 80     >422            ORA  #$80
EB14: 85 9E     >423            STA  FAC+1
EB16: 88        >424            DEY
EB17: B1 5E     >425            LDA  (INDEX),Y
EB19: 85 9D     >426            STA  FAC
EB1B: 84 AC     >427            STY  EXTRAFAC    ;Y=0
EB1D: 60        >428            RTS
                >429
EB1E: A2 98     >430  MOV2F     LDX  #TEMP2      ;Pack FAC into TEMP2
EB20: 2C        >431            HEX  2C          ;Trick to branch
EB21: A2 93     >432  MOV1F     LDX  #TEMP1      ;Pack FAC into TEMP1
EB23: A0 00     >433  MOVML     LDY  #0          ;High byte of dest adrs=0
EB25: F0 04     >434            BEQ  MOVMF
EB27: A6 85     >435  SETFOR    LDX  FORPNT
EB29: A4 86     >436            LDY  FORPNT+1
EB2B: 20 72 EB  >437  MOVMF     JSR  RNDB        ;Pack FAC into memory (X,Y)
EB2E: 86 5E     >438            STX  INDEX
EB30: 84 5F     >439            STY  INDEX+1
EB32: A0 04     >440            LDY  #4
EB34: A5 A1     >441            LDA  FAC+4
EB36: 91 5E     >442            STA  (INDEX),Y
EB38: 88        >443            DEY
EB39: A5 A0     >444            LDA  FAC+3
EB3B: 91 5E     >445            STA  (INDEX),Y
EB3D: 88        >446            DEY
EB3E: A5 9F     >447            LDA  FAC+2
EB40: 91 5E     >448            STA  (INDEX),Y
EB42: 88        >449            DEY
EB43: A5 A2     >450            LDA  FACSGN
EB45: 09 7F     >451            ORA  #$7F
EB47: 25 9E     >452            AND  FAC+1
```

```
EB49: 91 5E     >453            STA  (INDEX),Y
EB4B: 88        >454            DEY
EB4C: A5 9D     >455            LDA  FAC
EB4E: 91 5E     >456            STA  (INDEX),Y
  50: 84 AC     >457            STY  EXTRAFAC
EB52: 60        >458            RTS
                >459
EB53: A5 AA     >460  MOVFA     LDA  ARGSGN      ;Move ARG to FAC
EB55: 85 A2     >461  MFA       STA  FACSGN
EB57: A2 05     >462            LDX  #5
EB59: B5 A4     >463  MFA2      LDA  ARG-1,X
EB5B: 95 9C     >464            STA  FAC-1,X
EB5D: CA        >465            DEX
EB5E: D0 F9     >466            BNE  MFA2
EB60: 86 AC     >467            STX  EXTRAFAC
EB62: 60        >468            RTS
                >469
EB63: 20 72 EB  >470  MOVAF     JSR  RNDB        ;Round, then
EB66: A2 06     >471  MAF       LDX  #6          ;move FAC to ARG
EB68: B5 9C     >472  MAF2      LDA  FAC-1,X
EB6A: 95 A4     >473            STA  ARG-1,X
EB6C: CA        >474            DEX
EB6D: D0 F9     >475            BNE  MAF2
EB6F: 86 AC     >476            STX  EXTRAFAC
EB71: 60        >477  RTN9      RTS
                >478
                >479  * General purpose routine to round FAC using
                >480  * the most significant bit of EXTRAFAC:
                >481
 72: A5 9D      >482  RNDB      LDA  FAC
 74: F0 FB      >483            BEQ  RTN9
EB76: 06 AC     >484            ASL  EXTRAFAC    ;If EXTRAFAC is neg
EB78: 90 F7     >485            BCC  RTN9        ; then add one bit
EB7A: 20 C6 E8  >486  ROUND     JSR  PLUSEPS     ; to number in FAC.
EB7D: D0 F2     >487            BNE  RTN9
EB7F: 4C 8F E8  >488            JMP  FROUND      ;Round if exp affected
                >489
EB82: A5 9D     >490  SIGN      LDA  FAC         ;Check sign of FAC and
EB84: F0 09     >491            BEQ  RTN10       ; return -1,0,1 in A-reg
EB86: A5 A2     >492  SIGN1     LDA  FACSGN      ; according to result.
EB88: 2A        >493  SIGN2     ROL
EB89: A9 FF     >494            LDA  #$FF
EB8B: B0 02     >495            BCS  RTN10
EB8D: A9 01     >496            LDA  #1
EB8F: 60        >497  RTN10     RTS
                >498
EB90: 20 82 EB  >499  SGN       JSR  SIGN        ;Convert FAC to -1,0,1
EB93: 85 9E     >500  FLOAT     STA  FAC+1       ;Float signed contents
EB95: A9 00     >501            LDA  #0          ; of A-reg.
EB97: 85 9F     >502            STA  FAC+2
EB99: A2 88     >503            LDX  #$88
EB9B: A5 9E     >504  FLO1      LDA  FAC+1
EB9D: 49 FF     >505            EOR  #$FF
EB9F: 2A        >506            ROL
EBA0: A9 00     >507  FLO2      LDA  #0
  A2: 85 A1     >508            STA  FAC+4
EBA4: 85 A0     >509            STA  FAC+3
```

```
EBA6: 86 9D     >510            STX  FAC
EBA8: 85 AC     >511            STA  EXTRAFAC
EBAA: 85 A2     >512            STA  FACSGN
EBAC: 4C 29 E8  >513            JMP  SGNIF
                >514
EBAF: 46 A2     >515  ABS       LSR  FACSGN      ;Change sign to +
EBB1: 60        >516            RTS
                >517
                >518  * Routine to compare FAC with packed # at (Y,A):
                >519
EBB2: 85 60     >520  FCOMP     STA  DEST
EBB4: 84 61     >521  FCOMP2    STY  DEST+1
EBB6: A0 00     >522            LDY  #0
EBB8: B1 60     >523            LDA  (DEST),Y
EBBA: C8        >524            INY
EBBB: AA        >525            TAX
EBBC: F0 C4     >526            BEQ  SIGN
EBBE: B1 60     >527            LDA  (DEST),Y
EBC0: 45 A2     >528            EOR  FACSGN
EBC2: 30 C2     >529            BMI  SIGN1
EBC4: E4 9D     >530            CPX  FAC
EBC6: D0 21     >531            BNE  FC1
EBC8: B1 60     >532            LDA  (DEST),Y
EBCA: 09 80     >533            ORA  #$80
EBCC: C5 9E     >534            CMP  FAC+1
EBCE: D0 19     >535            BNE  FC1
EBD0: C8        >536            INY
EBD1: B1 60     >537            LDA  (DEST),Y
EBD3: C5 9F     >538            CMP  FAC+2
EBD5: D0 12     >539            BNE  FC1
EBD7: C8        >540            INY
EBD8: B1 60     >541            LDA  (DEST),Y
EBDA: C5 A0     >542            CMP  FAC+3
EBDC: D0 0B     >543            BNE  FC1
EBDE: C8        >544            INY
EBDF: A9 7F     >545            LDA  #$7F
EBE1: C5 AC     >546            CMP  EXTRAFAC
EBE3: B1 60     >547            LDA  (DEST),Y
EBE5: E5 A1     >548            SBC  FAC+4
EBE7: F0 28     >549            BEQ  RTN11       ;Exit if #s =
EBE9: A5 A2     >550  FC1       LDA  FACSGN
EBEB: 90 02     >551            BCC  FC2
EBED: 49 FF     >552            EOR  #$FF
EBEF: 4C 88 EB  >553  FC2       JMP  SIGN2
                >554
                >555  * On exit from FCOMP, A=1,0,-1 as (A,Y) is <,=,> FAC.
                >556
EBF2: A5 9D     >557  QINT      LDA  FAC         ;Convert FAC to
EBF4: F0 4A     >558            BEQ  ZFAC        ; its integer part.
EBF6: 38        >559            SEC              ; Assumes FAC < 2^31.
EBF7: E9 A0     >560            SBC  #$A0        ; Result is left in
EBF9: 24 A2     >561            BIT  FACSGN      ; FAC+1 to FAC+4
EBFB: 10 09     >562            BPL  QI1         ; ($9E-$A2).
EBFD: AA        >563            TAX
EBFE: A9 FF     >564            LDA  #$FF
EC00: 85 A4     >565            STA  FPGEN
EC02: 20 A4 E8  >566            JSR  NEG2
```

```
EC05: 8A        >567             TXA
EC06: A2 9D     >568  QI1        LDX  #FAC
EC08: C9 F9     >569             CMP  #$F9        ;More than 7 bits to shift?
EC0A: 10 06     >570             BPL  QI2         ;Branch if not
EC0C: 20 F0 E8  >571             JSR  SHIFT       ;Do byte shift if so
EC0F: 84 A4     >572             STY  FPGEN       ;Y=0
EC11: 60        >573  RTN11      RTS
                >574
EC12: A8        >575  QI2        TAY
EC13: A5 A2     >576             LDA  FACSGN
EC15: 29 80     >577             AND  #$80
EC17: 46 9E     >578             LSR  FAC+1
EC19: 05 9E     >579             ORA  FAC+1
EC1B: 85 9E     >580             STA  FAC+1
EC1D: 20 07 E9  >581             JSR  SHFTR
EC20: 84 A4     >582             STY  FPGEN
EC22: 60        >583             RTS
                >584
EC23: A5 9D     >585  INT        LDA  FAC
EC25: C9 A0     >586             CMP  #$A0        ;< 2^31 ?
EC27: B0 20     >587             BCS  RTN12       ;Exit if not
EC29: 20 F2 EB  >588             JSR  QINT
EC2C: 84 AC     >589             STY  EXTRAFAC    ;Y=0
EC2E: A5 A2     >590             LDA  FACSGN
EC30: 84 A2     >591             STY  FACSGN
EC32: 49 80     >592             EOR  #$80
EC34: 2A        >593             ROL
EC35: A9 A0     >594             LDA  #$A0
EC37: 85 9D     >595             STA  FAC
EC39: A5 A1     >596             LDA  FAC+4
EC3B: 85 0D     >597             STA  CHARAC
EC3D: 4C 29 E8  >598             JMP  SGNIF
                >599
EC40: 85 9E     >600  ZFAC       STA  FAC+1
EC42: 85 9F     >601             STA  FAC+2
EC44: 85 A0     >602             STA  FAC+3
EC46: 85 A1     >603             STA  FAC+4
EC48: A8        >604             TAY
EC49: 60        >605  RTN12      RTS
                >606
                >607  * Evaluate floating point number at TXTPTR:
                >608
EC4A: A0 00     >609  FIN        LDY  #0
EC4C: A2 0A     >610             LDX  #10         ;Zero TMPEXP to SERLEN
EC4E: 94 99     >611  FIN2       STY  TMPEXP,X    ;($99-$A3)
EC50: CA        >612             DEX
EC51: 10 FB     >613             BPL  FIN2
EC53: 90 0F     >614             BCC  NXDIGIT
EC55: C9 2D     >615             CMP  #'-'
EC57: D0 04     >616             BNE  FIN3
EC59: 86 A3     >617             STX  SERLEN      ;Flags neg num if -
EC5B: F0 04     >618             BEQ  EVAL
EC5D: C9 2B     >619  FIN3       CMP  #'+'
EC5F: D0 05     >620             BNE  CHKDP
EC61: 20 B1 00  >621  EVAL       JSR  CHRGET
EC64: 90 5B     >622  NXDIGIT    BCC  INSRTDIG
EC66: C9 2E     >623  CHKDP      CMP  #'.'
```

```
EC68: F0 2E     >624            BEQ  SETDP
EC6A: C9 45     >625            CMP  #'E'
EC6C: D0 30     >626            BNE  ADJEXP
EC6E: 20 B1 00  >627            JSR  CHRGET     ;Get exponent
  71: 90 17     >628            BCC  GOGEX
EC73: C9 C9     >629            CMP  #minus     ;Neg exp?
EC75: F0 0E     >630            BEQ  SETSGN     ;Flag if so
EC77: C9 2D     >631            CMP  #'-'       ;May not be in token form
EC79: F0 0A     >632            BEQ  SETSGN
EC7B: C9 C8     >633            CMP  #plus      ;Similarly for +
EC7D: F0 08     >634            BEQ  DPDIG
EC7F: C9 2B     >635            CMP  #'+'
EC81: F0 04     >636            BEQ  DPDIG
EC83: D0 07     >637            BNE  SGNCHK     ;Number completed
                >638
EC85: 66 9C     >639  SETSGN    ROR  EXPSGN     ;Flag neg exp
EC87: 20 B1 00  >640  DPDIG     JSR  CHRGET     ;Get next exp digit
EC8A: 90 5C     >641  GOGEX     BCC  GETEXP     ;Branch if number
EC8C: 24 9C     >642  SGNCHK    BIT  EXPSGN
EC8E: 10 0E     >643            BPL  ADJEXP
EC90: A9 00     >644            LDA  #0
EC92: 38        >645            SEC             ;Negate exponent
EC93: E5 9A     >646            SBC  EXPON
EC95: 4C A0 EC  >647            JMP  AEX
EC98: 66 9B     >648  SETDP     ROR  DPFLG
EC9A: 24 9B     >649            BIT  DPFLG
EC9C: 50 C3     >650            BVC  EVAL       ;Branch if first "."
                >651
                >652  * Appears that there should be a jump to error here.
                >653  * In fact, multiple decimal points give strange results
                >654  * in PRINT statements.  Variable assignments correctly
                >655  * give syntax errors.
                >656
EC9E: A5 9A     >657  ADJEXP   LDA  EXPON      ;Adjust the exponent and exit
ECA0: 38        >658  AEX      SEC
ECA1: E5 99     >659           SBC  TMPEXP
ECA3: 85 9A     >660           STA  EXPON
ECA5: F0 12     >661           BEQ  EVDONE
ECA7: 10 09     >662           BPL  DPRIGHT
ECA9: 20 55 EA  >663  DPLEFT   JSR  DIV10
ECAC: E6 9A     >664           INC  EXPON
ECAE: D0 F9     >665           BNE  DPLEFT
ECB0: F0 07     >666           BEQ  EVDONE
ECB2: 20 39 EA  >667  DPRIGHT  JSR  MUL10
ECB5: C6 9A     >668           DEC  EXPON
ECB7: D0 F9     >669           BNE  DPRIGHT
ECB9: A5 A3     >670  EVDONE   LDA  SERLEN     ;Negative?
ECBB: 30 01     >671           BMI  EVD
ECBD: 60        >672           RTS
ECBE: 4C D0 EE  >673  EVD      JMP  NEGOP
                >674
ECC1: 48        >675  INSRTDIG PHA             ;Save digit
ECC2: 24 9B     >676           BIT  DPFLG      ;Was there a decimal pnt?
ECC4: 10 02     >677           BPL  NDP        ;Branch if not
ECC6: E6 99     >678           INC  TMPEXP     ;Adjust if so
  C8: 20 39 EA  >679  NDP      JSR  MUL10      ;Dec pnt over
ECCB: 68        >680           PLA             ;Add digit to left of dp
```

```
ECCC: 38        >681            SEC
ECCD: E9 30     >682            SBC  #'0'        ;Mask
ECCF: 20 D5 EC  >683            JSR  ADDACC
ECD2: 4C 61 EC  >684            JMP  EVAL        ;Loop until done
                >685
                >686  * Routine to add A-register to FAC:
                >687
ECD5: 48        >688  ADDACC    PHA
ECD6: 20 63 EB  >689            JSR  MOVAF       ;Copy FAC to ARG
ECD9: 68        >690            PLA
ECDA: 20 93 EB  >691            JSR  FLOAT
ECDD: A5 AA     >692            LDA  ARGSGN
ECDF: 45 A2     >693            EOR  FACSGN
ECE1: 85 AB     >694            STA  SGNCPR
ECE3: A6 9D     >695            LDX  FAC         ;To signal if FAC=0
ECE5: 4C C1 E7  >696            JMP  FADDT
                >697
ECE8: A5 9A     >698  GETEXP    LDA  EXPON
ECEA: C9 0A     >699            CMP  #10
ECEC: 90 09     >700            BCC  MVDG
ECEE: A9 64     >701            LDA  #100        ;Too big
ECF0: 24 9C     >702            BIT  EXPSGN      ;Is exp neg?
ECF2: 30 11     >703            BMI  STEX        ;If so will get 0
ECF4: 4C D5 E8  >704            JMP  OVERFLOW    ;If not, overflow
ECF7: 0A        >705  MVDG      ASL              ;times 10
ECF8: 0A        >706            ASL
ECF9: 18        >707            CLC
ECFA: 65 9A     >708            ADC  EXPON
ECFC: 0A        >709            ASL
ECFD: 18        >710            CLC
ECFE: A0 00     >711            LDY  #0
ED00: 71 B8     >712            ADC  (TXTPTR),Y
ED02: 38        >713            SEC
ED03: E9 30     >714            SBC  #'0'
ED05: 85 9A     >715  STEX      STA  EXPON
ED07: 4C 87 EC  >716            JMP  DPDIG
                >717
ED0A: 9B 3E BC
ED0D: 1F FD     >718  HMmiTNTH  HEX  9B3EBC1FFD ;99,999,999.9
ED0F: 9E 6E 6B
ED12: 27 FD     >719  BILmiONE  HEX  9E6E6B27FD ;999,999,999
ED14: 9E 6E 6B
ED17: 28 00     >720  BILLION   HEX  9E6E6B2800 ;1,000,000,000
                >721
ED19: A9 58     >722  INPRT     LDA  #INMSG      ;Print " IN "
ED1B: A0 D3     >723            LDY  #>INMSG
ED1D: 20 31 ED  >724            JSR  PRSTR
ED20: A5 76     >725            LDA  CURLIN+1
ED22: A6 75     >726            LDX  CURLIN
ED24: 85 9E     >727  LINPRT    STA  FAC+1       ;Print A,X in decimal
ED26: 86 9F     >728            STX  FAC+2
ED28: A2 90     >729            LDX  #$90
ED2A: 38        >730            SEC
ED2B: 20 A0 EB  >731            JSR  FLO2
ED2E: 20 34 ED  >732  PRNTFAC   JSR  FOUT        ;Print FP # in FAC
  31: 4C 3A DB  >733  PRSTR     JMP  STROUT      ;Print string at A,Y
                >734
```

```
                    >735  * Convert FAC to a string at STACK and point
                    >736  * A,Y to it:
                    >737
ED34: A0 01         >738  FOUT     LDY  #1
                    >739
                    >740  * Entry from STR$ routine puts string at $FF (Y=0)
                    >741  * so as to use an indirect address to point to it:
                    >742
ED36: A9 2D         >743  FACSTRNG LDA  #'-'
ED38: 88            >744          DEY
ED39: 24 A2         >745          BIT  FACSGN
ED3B: 10 04         >746          BPL  SFSG
ED3D: C8            >747          INY
ED3E: 99 FF 00      >748          STA  STACK-1,Y
ED41: 85 A2         >749  SFSG     STA  FACSGN
ED43: 84 AD         >750          STY  STRNG2
ED45: C8            >751          INY
ED46: A9 30         >752          LDA  #'0'
ED48: A6 9D         >753          LDX  FAC          ;Number=0?
ED4A: D0 03         >754          BNE  NOTZE
ED4C: 4C 57 EE      >755          JMP  WNDUP        ;Finish up if so
ED4F: A9 00         >756  NOTZE    LDA  #0
ED51: E0 80         >757          CPX  #$80         ;Number>=1?
ED53: F0 02         >758          BEQ  MB
ED55: B0 09         >759          BCS  STE          ;Skip if not >= 1
ED57: A9 14         >760  MB       LDA  #BILLION
ED59: A0 ED         >761          LDY  #>BILLION
ED5B: 20 7F E9      >762          JSR  FMULT        ;Move dec pnt and
ED5E: A9 F7         >763          LDA  #$F7         ; fix exp
ED60: 85 99         >764  STE      STA  TMPEXP
ED62: A9 0F         >765  CMPBM1   LDA  #BILmiONE
ED64: A0 ED         >766          LDY  #>BILmiONE
ED66: 20 B2 EB      >767          JSR  FCOMP        ;Normalize between
ED69: F0 1E         >768          BEQ  INTPART      ; 100,000,000 and
ED6B: 10 12         >769          BPL  JD10         ; 999,999,999
ED6D: A9 0A         >770  CMPHM    LDA  #HMmiTNTH
ED6F: A0 ED         >771          LDY  #>HMmiTNTH
ED71: 20 B2 EB      >772          JSR  FCOMP
ED74: F0 02         >773          BEQ  JM10
ED76: 10 0E         >774          BPL  ROUN
ED78: 20 39 EA      >775  JM10     JSR  MUL10
ED7B: C6 99         >776          DEC  TMPEXP
ED7D: D0 EE         >777          BNE  CMPHM
ED7F: 20 55 EA      >778  JD10     JSR  DIV10
ED82: E6 99         >779          INC  TMPEXP
ED84: D0 DC         >780          BNE  CMPBM1
ED86: 20 A0 E7      >781  ROUN     JSR  FADDH        ;Round it
ED89: 20 F2 EB      >782  INTPART  JSR  QINT         ;Convert normal form to int
ED8C: A2 01         >783          LDX  #1
ED8E: A5 99         >784          LDA  TMPEXP
ED90: 18            >785          CLC
ED91: 69 0A         >786          ADC  #10
ED93: 30 09         >787          BMI  DPLOC
ED95: C9 0B         >788          CMP  #11
ED97: B0 06         >789          BCS  DPL
ED99: 69 FF         >790          ADC  #$FF
ED9B: AA            >791          TAX
```

```
ED9C: A9 02     >792            LDA  #2
ED9E: 38        >793  DPLOC     SEC                 ;Where does . go?
ED9F: E9 02     >794  DPL       SBC  #2
EDA1: 85 9A     >795            STA  EXPON
  A3: 86 99     >796            STX  TMPEXP
EDA5: 8A        >797            TXA
EDA6: F0 02     >798            BEQ  PUTDP
EDA8: 10 13     >799            BPL  MAKSTR
EDAA: A4 AD     >800  PUTDP     LDY  STRNG2
EDAC: A9 2E     >801            LDA  #'.'
EDAE: C8        >802            INY
EDAF: 99 FF 00  >803            STA  STACK-1,Y
EDB2: 8A        >804            TXA
EDB3: F0 06     >805            BEQ  SVY
EDB5: A9 30     >806            LDA  #'0'
EDB7: C8        >807            INY
EDB8: 99 FF 00  >808            STA  STACK-1,Y
EDBB: 84 AD     >809  SVY       STY  STRNG2
EDBD: A0 00     >810  MAKSTR    LDY  #0             ;Zero in on # while
EDBF: A2 80     >811            LDX  #$80           ; building string.
EDC1: A5 A1     >812  MSLUP     LDA  FAC+4
EDC3: 18        >813            CLC
EDC4: 79 6C EE  >814            ADC  DECTBL+3,Y
EDC7: 85 A1     >815            STA  FAC+4
EDC9: A5 A0     >816            LDA  FAC+3
EDCB: 79 6B EE  >817            ADC  DECTBL+2,Y
EDCE: 85 A0     >818            STA  FAC+3
EDD0: A5 9F     >819            LDA  FAC+2
EDD2: 79 6A EE  >820            ADC  DECTBL+1,Y
  D5: 85 9F     >821            STA  FAC+2
  D7: A5 9E     >822            LDA  FAC+1
EDD9: 79 69 EE  >823            ADC  DECTBL,Y
EDDC: 85 9E     >824            STA  FAC+1
EDDE: E8        >825            INX                 ;Count in X
EDDF: B0 04     >826            BCS  PARITY?
EDE1: 10 DE     >827            BPL  MSLUP
EDE3: 30 02     >828            BMI  COUNTED
EDE5: 30 DA     >829  PARITY?   BMI  MSLUP
EDE7: 8A        >830  COUNTED   TXA
EDE8: 90 04     >831            BCC  MAKDIGIT
EDEA: 49 FF     >832            EOR  #$FF
EDEC: 69 0A     >833            ADC  #10
EDEE: 69 2F     >834  MAKDIGIT  ADC  #'0'-1
EDF0: C8        >836            INY
EDF1: C8        >836            INY
EDF2: C8        >836            INY
EDF3: C8        >836            INY
EDF4: 84 83     >838            STY  VARPNT         ;Save ptr to DECTBL
EDF6: A4 AD     >839            LDY  STRNG2         ;Get ptr to string
EDF8: C8        >840            INY
EDF9: AA        >841            TAX
EDFA: 29 7F     >842            AND  #$7F
EDFC: 99 FF 00  >843            STA  STACK-1,Y
EDFF: C6 99     >844            DEC  TMPEXP
EE01: D0 06     >845            BNE  SAVY
  03: A9 2E     >846            LDA  #'.'
EE05: C8        >847            INY
```

```
EE06: 99 FF 00 >848              STA  STACK-1,Y
EE09: 84 AD    >849  SAVY        STY  STRNG2      ;Save string ptr
EE0B: A4 83    >850              LDY  VARPNT      ;Get DECTBL ptr
EE0D: 8A       >851              TXA
  0E: 49 FF    >852              EOR  #$FF
EE10: 29 80    >853              AND  #$80
EE12: AA       >854              TAX
EE13: C0 24    >855              CPY  #TEND-DECTBL
EE15: D0 AA    >856              BNE  MSLUP       ;Loop till done
EE17: A4 AD    >857              LDY  STRNG2
EE19: B9 FF 00 >858  MVBACK      LDA  STACK-1,Y
EE1C: 88       >859              DEY
EE1D: C9 30    >860              CMP  #'0'        ;Suppress trailing 0's
EE1F: F0 F8    >861              BEQ  MVBACK
EE21: C9 2E    >862              CMP  #'.'        ;If ends in DP, write over it
EE23: F0 01    >863              BEQ  NEEDEX?
EE25: C8       >864              INY
EE26: A9 2B    >865  NEEDEX?     LDA  #'+'
EE28: A6 9A    >866              LDX  EXPON
EE2A: F0 2E    >867              BEQ  MARKEND     ;Branch if no exp
EE2C: 10 08    >868              BPL  PUTEX
EE2E: A9 00    >869              LDA  #0
EE30: 38       >870              SEC
EE31: E5 9A    >871              SBC  EXPON
EE33: AA       >872              TAX
EE34: A9 2D    >873              LDA  #'-'
EE36: 99 01 01 >874  PUTEX       STA  STACK+1,Y
EE39: A9 45    >875              LDA  #'E'
EE3B: 99 00 01 >876              STA  STACK,Y
  3E: 8A       >877              TXA
  3F: A2 2F    >878              LDX  #'0'-1
EE41: 38       >879              SEC
EE42: E8       >880  WHATX       INX
EE43: E9 0A    >881              SBC  #10
EE45: B0 FB    >882              BCS  WHATX
EE47: 69 3A    >883              ADC  #'0'+10
EE49: 99 03 01 >884              STA  STACK+3,Y
EE4C: 8A       >885              TXA
EE4D: 99 02 01 >886              STA  STACK+2,Y
EE50: A9 00    >887              LDA  #0
EE52: 99 04 01 >888              STA  STACK+4,Y
EE55: F0 08    >889              BEQ  PNTSTK
EE57: 99 FF 00 >890  WNDUP       STA  STACK-1,Y
EE5A: A9 00    >891  MARKEND     LDA  #0
EE5C: 99 00 01 >892              STA  STACK,Y
EE5F: A9 00    >893  PNTSTK      LDA  #<STACK
EE61: A0 01    >894              LDY  #>STACK
EE63: 60       >895              RTS
               >896
EE64: 80 00 00
EE67: 00 00    >897  HALF        HEX  8000000000
               >898
               >899  * 32 bit hex reps of powers of 10:
               >900
EE69: FA 0A 1F
  6C: 00       >901  DECTBL      HEX  FA0A1F00    ;-100000000
EE6D: 00 98 96
```

```
EE70: 80        >902            HEX   00989680    ;10000000
EE71: FF F0 BD
EE74: C0        >903            HEX   FFF0BDC0    ;-1000000
EE75: 00 01 86
EE78: A0        >904            HEX   000186A0    ;100000
EE79: FF FF D8
EE7C: F0        >905            HEX   FFFFD8F0    ;-10000
EE7D: 00 00 03
EE80: E8        >906            HEX   000003E8    ;1000
EE81: FF FF FF
EE84: 9C        >907            HEX   FFFFFF9C    ;-100
EE85: 00 00 00
EE88: 0A        >908            HEX   0000000A    ;10
EE89: FF FF FF
EE8C: FF        >909            HEX   FFFFFFFF    ;-1
                >910
                >911  TEND      =     *
                >912
EE8D: 20 63 EB  >913  SQR       JSR   MOVAF       ;Compute as 1/2 power
EE90: A9 64     >914            LDA   #HALF
EE92: A0 EE     >915            LDY   #>HALF
EE94: 20 F9 EA  >916            JSR   MOVFM       ;Put 1/2 in FAC
                >917
EE97: F0 70     >918  FPWRT     BEQ   EXP         ;ARG^FAC -> FAC
EE99: A5 A5     >919            LDA   ARG
EE9B: D0 03     >920            BNE   PW1
EE9D: 4C 50 E8  >921            JMP   AtoFAC      ;Set FAC=0 if ARG=0
                >922
EEA0: A2 8A     >923  PW1       LDX   #TEMP3
EEA2: A0 00     >924            LDY   #0
EEA4: 20 2B EB  >925            JSR   MOVMF       ;Store at TEMP3
EEA7: A5 AA     >926            LDA   ARGSGN
EEA9: 10 0F     >927            BPL   PW2         ;Branch if argument is +
EEAB: 20 23 EC  >928            JSR   INT         ;Get INT part of exponent
EEAE: A9 8A     >929            LDA   #TEMP3
EEB0: A0 00     >930            LDY   #0
EEB2: 20 B2 EB  >931            JSR   FCOMP       ;Is it an integer power?
EEB5: D0 03     >932            BNE   PW2
EEB7: 98        >933            TYA               ;If so, allow neg argument
EEB8: A4 0D     >934            LDY   CHARAC
EEBA: 20 55 EB  >935  PW2       JSR   MFA         ;Move argument to FAC
EEBD: 98        >936            TYA
EEBE: 48        >937            PHA
EEBF: 20 41 E9  >938            JSR   LOG         ;Get LOG(argument)
EEC2: A9 8A     >939            LDA   #TEMP3
EEC4: A0 00     >940            LDY   #0
EEC6: 20 7F E9  >941            JSR   FMULT       ;Compute expon*LOG(argum)
EEC9: 20 09 EF  >942            JSR   EXP         ;Raise to e-th power
EECC: 68        >943            PLA
EECD: 4A        >944            LSR               ;Was exponent a - integer?
EECE: 90 0A     >945            BCC   RTN13       ;Return if not
EED0: A5 9D     >946  NEGOP     LDA   FAC         ;Is result 0?
EED2: F0 06     >947            BEQ   RTN13       ;Return if so
                >948            >>>   NEG.FACSGN
EED4: A5 A2     >948            LDA   FACSGN
EED6: 49 FF     >948            EOR   #$FF
EED8: 85 A2     >948            STA   FACSGN
```

```
                        >948            <<<
EEDA: 60                >949  RTN13     RTS
                        >950
EEDB: 81 38 AA
  DE: 3B 29             >951  LOGe      HEX   8138AA3B29 ;LOG(e) to base 2
EEE0: 07                >952  EXPSER    DFB   7           ;Index to # of coefs:
EEE1: 71 34 58
EEE4: 3E 56             >953            HEX   7134583E56 ;(log(2)^7)/8!
EEE6: 74 16 7E
EEE9: B3 1B             >954            HEX   74167EB31B ;(log(2)^6)/7!
EEEB: 77 2F EE
EEEE: E3 85             >955            HEX   772FEEE385 ;(log(2)^5)/6!
EEF0: 7A 1D 84
EEF3: 1C 2A             >956            HEX   7A1D841C2A ;(log(2)^4)/5!
EEF5: 7C 63 59
EEF8: 58 0A             >957            HEX   7C6359580A ;(log(2)^3)/4!
EEFA: 7E 75 FD
EEFD: E7 C6             >958            HEX   7E75FDE7C6 ;(log(2)^2)/3!
EEFF: 80 31 72
EF02: 18 10             >959            HEX   8031721810 ;log(2)/2!
EF04: 81 00 00
EF07: 00 00             >960            HEX   8100000000 ;1
                        >961
EF09: A9 DB             >962  EXP       LDA   #LOGe       ; e^FAC -> FAC
EF0B: A0 EE             >963            LDY   #>LOGe
EF0D: 20 7F E9          >964            JSR   FMULT       ;Set up to compute as 2^(xLOG(e))
EF10: A5 AC             >965            LDA   EXTRAFAC
EF12: 69 50             >966            ADC   #$50
EF14: 90 03             >967            BCC   X1
 16: 20 7A EB           >968            JSR   ROUND
 19: 85 92              >969  X1        STA   EXTRASV
EF1B: 20 66 EB          >970            JSR   MAF         ;Copy to ARG
EF1E: A5 9D             >971            LDA   FAC
EF20: C9 88             >972            CMP   #$88        ;Within range?
EF22: 90 03             >973            BCC   X2          ;Branch if not
EF24: 20 2B EA          >974  OOR       JSR   OUTOFRNG    ;Make zero or overflow
EF27: 20 23 EC          >975  X2        JSR   INT         ;Get integer part in FAC
EF2A: A5 0D             >976            LDA   CHARAC
EF2C: 18                >977            CLC
EF2D: 69 81             >978            ADC   #$81
EF2F: F0 F3             >979            BEQ   OOR
EF31: 38                >980            SEC
EF32: E9 01             >981            SBC   #1
EF34: 48                >982            PHA
EF35: A2 05             >983            LDX   #5
EF37: B5 A5             >984  X3        LDA   ARG,X       ;Swap ARG and FAC
EF39: B4 9D             >985            LDY   FAC,X
EF3B: 95 9D             >986            STA   FAC,X
EF3D: 94 A5             >987            STY   ARG,X
EF3F: CA                >988            DEX
EF40: 10 F5             >989            BPL   X3
EF42: A5 92             >990            LDA   EXTRASV
EF44: 85 AC             >991            STA   EXTRAFAC
EF46: 20 AA E7          >992            JSR   FSUBT       ;Subtract off integer part
EF49: 20 D0 EE          >993            JSR   NEGOP
  4C: A9 E0             >994            LDA   #EXPSER
EF4E: A0 EE             >995            LDY   #>EXPSER
```

```
EF50: 20 72 EF >996            JSR  SERIES      ;Use series on frac. part
EF53: A9 00    >997            LDA  #0
EF55: 85 AB    >998            STA  SGNCPR
EF57: 68       >999            PLA
   58: 20 10 EA >1000           JSR  ADEX2       ;Add exponent of int. part
EF5B: 60       >1001           RTS
               >1002
EF5C: 85 AD    >1003 ODDSER    STA  SERPNT      ;Computes ax+bx^3+cx^5+...
EF5E: 84 AE    >1004           STY  SERPNT+1    ; where SERPNT points to
EF60: 20 21 EB >1005           JSR  MOV1F       ; coef ...c,b,a.
EF63: A9 93    >1006           LDA  #TEMP1      ; # of coef = SERLEN+1
EF65: 20 7F E9 >1007           JSR  FMULT       ;Square x
EF68: 20 76 EF >1008           JSR  SERMAIN     ;Do series in x^2
EF6B: A9 93    >1009           LDA  #TEMP1      ;Get x again
EF6D: A0 00    >1010           LDY  #0
EF6F: 4C 7F E9 >1011           JMP  FMULT       ;Multiply by series and exit
               >1012
EF72: 85 AD    >1013 SERIES    STA  SERPNT      ;Computes a+bx+cx^2+...
EF74: 84 AE    >1014           STY  SERPNT+1    ; where SERPNT points to
EF76: 20 1E EB >1015 SERMAIN   JSR  MOV2F       ; coef ...c,b,a.
EF79: B1 AD    >1016           LDA  (SERPNT),Y
EF7B: 85 A3    >1017           STA  SERLEN      ;Set up SERLEN from table start
EF7D: A4 AD    >1018           LDY  SERPNT      ;and point SERPNT to last coef
EF7F: C8       >1019           INY              ;(which comes first in table).
EF80: 98       >1020           TYA
EF81: D0 02    >1021           BNE  SS
EF83: E6 AE    >1022           INC  SERPNT+1
EF85: 85 AD    >1023 SS        STA  SERPNT
EF87: A4 AE    >1024           LDY  SERPNT+1
   89: 20 7F E9 >1025 SERLOOP  JSR  FMULT
   8C: A5 AD    >1026           LDA  SERPNT
EF8E: A4 AE    >1027           LDY  SERPNT+1
EF90: 18       >1028           CLC
EF91: 69 05    >1029           ADC  #5          ;Move SERPNT to next coef
EF93: 90 01    >1030           BCC  NXTERM
EF95: C8       >1031           INY
EF96: 85 AD    >1032 NXTERM    STA  SERPNT
EF98: 84 AE    >1033           STY  SERPNT+1
EF9A: 20 BE E7 >1034           JSR  FADD        ;Add one term
EF9D: A9 98    >1035           LDA  #TEMP2      ;Get x again
EF9F: A0 00    >1036           LDY  #0
EFA1: C6 A3    >1037           DEC  SERLEN
EFA3: D0 E4    >1038           BNE  SERLOOP     ;Loop till done
EFA5: 60       >1039 RTN14     RTS
               >1040
EFA6: 98 35 44
EFA9: 7A       >1041 RNDADJ1   HEX  9835447A
EFAA: 68 28 B1
EFAD: 46       >1042 RNDADJ2   HEX  6828B146
               >1043
EFAE: 20 82 EB >1044 RND       JSR  SIGN        ;Get sign of argument
EFB1: AA       >1045           TAX              ;Remember it
EFB2: 30 18    >1046           BMI  RD1         ;If - use current FAC
EFB4: A9 C9    >1047           LDA  #RNDSEED
EFB6: A0 00    >1048           LDY  #0          ;Move current seed to FAC
   B8: 20 F9 EA >1049          JSR  MOVFM
EFBB: 8A       >1050           TXA              ;Recall sign
```

```
EFBC: F0 E7     >1051          BEQ  RTN14       ;Exit now if RND(0)
EFBE: A9 A6     >1052          LDA  #RNDADJ1    ;Mix it up
EFC0: A0 EF     >1053          LDY  #>RNDADJ1
EFC2: 20 7F E9  >1054          JSR  FMULT
  C5: A9 AA     >1055          LDA  #RNDADJ2    ;More mixing
EFC7: A0 EF     >1056          LDY  #>RNDADJ2
EFC9: 20 BE E7  >1057          JSR  FADD
EFCC: A6 A1     >1058 RD1      LDX  FAC+4       ;Still more
EFCE: A5 9E     >1059          LDA  FAC+1
EFD0: 85 A1     >1060          STA  FAC+4
EFD2: 86 9E     >1061          STX  FAC+1
EFD4: A9 00     >1062          LDA  #0
EFD6: 85 A2     >1063          STA  FACSGN
EFD8: A5 9D     >1064          LDA  FAC
EFDA: 85 AC     >1065          STA  EXTRAFAC
EFDC: A9 80     >1066          LDA  #$80
EFDE: 85 9D     >1067          STA  FAC
EFE0: 20 2E E8  >1068          JSR  SIGNIF
EFE3: A2 C9     >1069          LDX  #RNDSEED    ;Move FAC to rnd seed
EFE5: A0 00     >1070          LDY  #0
EFE7: 4C 2B EB  >1071 RD2      JMP  MOVMF
                >1072
EFEA: A9 66     >1073 COS      LDA  #PIhalf     ;Cos(x)=sin(x + pi/2)
EFEC: A0 F0     >1074          LDY  #>PIhalf
EFEE: 20 BE E7  >1075          JSR  FADD
                >1076
EFF1: 20 63 EB  >1077 SIN      JSR  MOVAF       ;Copy FAC to ARG
EFF4: A9 6B     >1078          LDA  #PIdoub
EFF6: A0 F0     >1079          LDY  #>PIdoub
 F8: A6 AA      >1080          LDX  ARGSGN
 FA: 20 5E EA   >1081          JSR  DIV         ;Divide by 2pi
EFFD: 20 63 EB  >1082          JSR  MOVAF       ;Copy to ARG
F000: 20 23 EC  >1083          JSR  INT         ;Take integer part
F003: A9 00     >1084          LDA  #0          ;Does nothing
F005: 85 AB     >1085          STA  SGNCPR      ; "
F007: 20 AA E7  >1086          JSR  FSUBT       ;Subtract to get mod(2pi)
F00A: A9 70     >1087          LDA  #QUARTER
F00C: A0 F0     >1088          LDY  #>QUARTER
F00E: 20 A7 E7  >1089          JSR  FSUB        ;Convert argument to 1st quad
F011: A5 A2     >1090          LDA  FACSGN      ; range 0 to 1/4 as
F013: 48        >1091          PHA              ; multiples of 2pi
F014: 10 0D     >1092          BPL  SI1
F016: 20 A0 E7  >1093          JSR  FADDH
F019: A5 A2     >1094          LDA  FACSGN
F01B: 30 09     >1095          BMI  SI2
                >1096          >>>  NEG.SIGNFLG
F01D: A5 16     >1096          LDA  SIGNFLG
F01F: 49 FF     >1096          EOR  #$FF
F021: 85 16     >1096          STA  SIGNFLG
                >1096          <<<
F023: 20 D0 EE  >1097 SI1      JSR  NEGOP
F026: A9 70     >1098 SI2      LDA  #QUARTER
F028: A0 F0     >1099          LDY  #>QUARTER
F02A: 20 BE E7  >1100          JSR  FADD
F02D: 68        >1101          PLA
  2E: 10 03     >1102          BPL  SI3
F030: 20 D0 EE  >1103          JSR  NEGOP
```

```
F033: A9 75     >1104 SI3       LDA  #SINSER     ;Do standard sin series
F035: A0 F0     >1105           LDY  #>SINSER
F037: 4C 5C EF  >1106           JMP  ODDSER
                >1107
 3A: 20 21 EB   >1108 TAN       JSR  MOV1F       ;Save FAC in TEMP1
F03D: A9 00     >1109           LDA  #0
F03F: 85 16     >1110           STA  SIGNFLG
F041: 20 F1 EF  >1111           JSR  SIN
F044: A2 8A     >1112           LDX  #TEMP3
F046: A0 00     >1113           LDY  #0          ;Store sin at TEMP3
F048: 20 E7 EF  >1114           JSR  RD2
F04B: A9 93     >1115           LDA  #TEMP1
F04D: A0 00     >1116           LDY  #0
F04F: 20 F9 EA  >1117           JSR  MOVFM       ;Retrieve FAC
F052: A9 00     >1118           LDA  #0          ; and compute cos
F054: 85 A2     >1119           STA  FACSGN
F056: A5 16     >1120           LDA  SIGNFLG
F058: 20 62 F0  >1121           JSR  TAN2
F05B: A9 8A     >1122           LDA  #TEMP3      ;Retieve sin
F05D: A0 00     >1123           LDY  #0
F05F: 4C 66 EA  >1124           JMP  FDIV        ; and divide
                >1125
F062: 48        >1126 TAN2      PHA
F063: 4C 23 F0  >1127           JMP  SI1
                >1128
F066: 81 49 0F
F069: DA A2     >1129 PIhalf    HEX  81490FDAA2
F06B: 83 49 0F
F06E: DA A2     >1130 PIdoub    HEX  83490FDAA2
 70: 7F 00 00
 73: 00 00      >1131 QUARTER   HEX  7F00000000
F075: 05        >1132 SINSER    DFB  5           ;Index to # of coefs:
F076: 84 E6 1A
F079: 2D 1B     >1133           HEX  84E61A2D1B  ;(2pi)^11/11!
F07B: 86 28 07
F07E: FB F8     >1134           HEX  862807FBF8  ;(2pi)^9/9!
F080: 87 99 68
F083: 89 01     >1135           HEX  8799688901  ;(2pi)^7/7!
F085: 87 23 35
F088: DF E1     >1136           HEX  872335DFE1  ;(2pi)^5/5!
F08A: 86 A5 5D
F08D: E7 28     >1137           HEX  86A55DE728  ;(2pi)^3/3!
F08F: 83 49 0F
F092: DA A2     >1138           HEX  83490FDAA2  ;2pi
F094: A6 D3 C1
F097: C8 D4     >1139           HEX  A6D3C1C8D4  ;Does not appear used
F099: C8 D5 C4
F09C: CE CA     >1140           HEX  C8D5C4CECA  ;"
                >1141
F09E: A5 A2     >1142 ATN       LDA  FACSGN      ;A modified Gregory series
F0A0: 48        >1143           PHA              ; is used here.  (Gregory
F0A1: 10 03     >1144           BPL  ATN1        ; converges too slowly)
F0A3: 20 D0 EE  >1145           JSR  NEGOP
F0A6: A5 9D     >1146 ATN1      LDA  FAC
F0A8: 48        >1147           PHA
 A9: C9 81      >1148           CMP  #$81        ;Normalize between 0 & 1
F0AB: 90 07     >1149           BCC  ATN2
```

```
F0AD: A9 13     >1150          LDA  #ONE
F0AF: A0 E9     >1151          LDY  #>ONE
F0B1: 20 66 EA  >1152          JSR  FDIV
F0B4: A9 CE     >1153 ATN2     LDA  #ATNSER
  B6: A0 F0     >1154          LDY  #>ATNSER
F0B8: 20 5C EF  >1155          JSR  ODDSER
F0BB: 68        >1156          PLA
F0BC: C9 81     >1157          CMP  #$81
F0BE: 90 07     >1158          BCC  ATN3
F0C0: A9 66     >1159          LDA  #PIhalf
F0C2: A0 F0     >1160          LDY  #>PIhalf
F0C4: 20 A7 E7  >1161          JSR  FSUB       ;Compensate for normalization
F0C7: 68        >1162 ATN3     PLA
F0C8: 10 03     >1163          BPL  RTN15
F0CA: 4C D0 EE  >1164          JMP  NEGOP
F0CD: 60        >1165 RTN15    RTS
                >1166
F0CE: 0B        >1167 ATNSER   DFB  11         ;Index to # of coefs:
F0CF: 76 B3 83
F0D2: BD D3     >1168          HEX  76B383BDD3
F0D4: 79 1E F4
F0D7: A6 F5     >1169          HEX  791EF4A6F5
F0D9: 7B 83 FC
F0DC: B0 10     >1170          HEX  7B83FCB010
F0DE: 7C 0C 1F
F0E1: 67 CA     >1171          HEX  7C0C1F67CA
F0E3: 7C DE 53
F0E6: CB C1     >1172          HEX  7CDE53CBC1
F0E8: 7D 14 64
  EB: 70 4C     >1173          HEX  7D1464704C
  ED: 7D B7 EA
F0F0: 51 7A     >1174          HEX  7DB7EA517A
F0F2: 7D 63 30
F0F5: 88 7E     >1175          HEX  7D6330887E
F0F7: 7E 92 44
F0FA: 99 3A     >1176          HEX  7E9244993A
F0FC: 7E 4C CC
F0FF: 91 C7     >1177          HEX  7E4CCC91C7
F101: 7F AA AA
F104: AA 13     >1178          HEX  7FAAAAAA13
F106: 81 00 00
F109: 00 00     >1179          HEX  8100000000
                >1180
                >1181 * CHRGET routine (and RND seed)
                >1182 * to be placed at $B1 on zero page.
                >1183
                >1184 ZPSTUFF  >>>  INCR.TXTPTR
F10B: E6 B8     >1184          INC  TXTPTR
F10D: D0 02     >1184          BNE  NI
F10F: E6 B9     >1184          INC  TXTPTR+1
                >1184 NI       <<<
F111: AD 60 EA  >1185          LDA  $EA60      ;Address of no importance
F114: C9 3A     >1186          CMP  #':'       ;Return carry set if not #
F116: B0 0A     >1187          BCS  RTN16      ;Z-flag set if ':' or eol
F118: C9 20     >1188          CMP  #' '       ;Skip spaces
  1A: F0 EF     >1189          BEQ  ZPSTUFF
F11C: 38        >1190          SEC
```

```
F11D: E9 30     >1191           SBC  #'0'         ;This code clears carry if
F11F: 38        >1192           SEC               ; numeric, sets it if not,
F120: E9 D0     >1193           SBC  #$100-'0'    ; and leaves A-reg as found
F122: 60        >1194 RTN16     RTS
                >1195
F123: 80 4F C7
F126: 52 58     >1196           HEX  804FC75258 ;Random number seed
                >1197
F128: A2 FF     >1198 COLDST    LDX  #$FF
F12A: 86 76     >1199           STX  CURLIN+1     ;Init direct mode
F12C: A2 FB     >1200           LDX  #$FB         ; and stack pointer.
F12E: 9A        >1201           TXS               ;Upper 4 bytes of stack used for
                >1202                             ; link and line # in line input.
F12F: A9 28     >1203           LDA  #COLDST
F131: A0 F1     >1204           LDY  #>COLDST
F133: 85 01     >1205           STA  GOWARM+1     ;Why? These changed later!
F135: 84 02     >1206           STY  GOWARM+2
F137: 85 04     >1207           STA  GOSTROUT+1
F139: 84 05     >1208           STY  GOSTROUT+2
F13B: 20 73 F2  >1209           JSR  NORMAL       ;Init normal text
F13E: A9 4C     >1210           LDA  #$4C         ;Set up jump locations
F140: 85 00     >1211           STA  GOWARM
F142: 85 03     >1212           STA  GOSTROUT
F144: 85 90     >1213           STA  JMPADRS
F146: 85 0A     >1214           STA  USR          ;USR adrs inited
F148: A9 99     >1215           LDA  #IQERR       ; to illegal quantity
F14A: A0 E1     >1216           LDY  #>IQERR      ; error routine.
F14C: 85 0B     >1217           STA  USR+1
F14E: 84 0C     >1218           STY  USR+2
F150: A2 1C     >1219           LDX  #$1C         ;Should be $1D?
F152: BD 0A F1  >1220 MVZP      LDA  ZPSTUFF-1,X
F155: 95 B0     >1221           STA  CHRGET-1,X
F157: 86 F1     >1222           STX  SPEEDZ       ;Init SPEED to 255 (SPEEDZ = 1)
F159: CA        >1223           DEX
F15A: D0 F6     >1224           BNE  MVZP
F15C: 86 F2     >1225           STX  TRCFLG       ;Set NOTRACE
F15E: 8A        >1226           TXA
F15F: 85 A4     >1227           STA  FPGEN
F161: 85 54     >1228           STA  LASTPT+1
F163: 48        >1229           PHA               ;Put 0 at $1FB, SP now $FA
F164: A9 03     >1230           LDA  #3           ;This seems to have
F166: 85 8F     >1231           STA  DSCLEN       ; no purpose here.
F168: 20 FB DA  >1232           JSR  CRDO
F16B: A9 01     >1233           LDA  #1           ;Set up fake
F16D: 8D FD 01  >1234           STA  IN-3         ; link of $101
F170: 8D FC 01  >1235           STA  IN-4
F173: A2 55     >1236           LDX  #$55         ;Init index to temp
F175: 86 52     >1237           STX  TEMPPT       ; string descriptors
F177: A9 00     >1238           LDA  #0
F179: A0 08     >1239           LDY  #8
F17B: 85 50     >1240           STA  LINNUM
F17D: 84 51     >1241           STY  LINNUM+1
F17F: A0 00     >1242           LDY  #0
F181: E6 51     >1243 FNDMEMHI  INC  LINNUM+1     ;Test first byte of each page
F183: B1 50     >1244           LDA  (LINNUM),Y
F185: 49 FF     >1245           EOR  #$FF         ; until ROM or empty location
F187: 91 50     >1246           STA  (LINNUM),Y   ; is found.
```

```
F189: D1 50     >1247           CMP  (LINNUM),Y
F18B: D0 08     >1248           BNE  MEMFOUND
F18D: 49 FF     >1249           EOR  #$FF         ;Put back as found
F18F: 91 50     >1250           STA  (LINNUM),Y
  91: D1 50     >1251           CMP  (LINNUM),Y ;Test again to make sure
F193: F0 EC     >1252           BEQ  FNDMEMHI
F195: A4 50     >1253 MEMFOUND  LDY  LINNUM
F197: A5 51     >1254           LDA  LINNUM+1
F199: 29 F0     >1255           AND  #$F0         ;Make sure it is a multiple
F19B: 84 73     >1256           STY  MEMSIZ       ; of 4K in case test faulty.
F19D: 85 74     >1257           STA  MEMSIZ+1
F19F: 84 6F     >1258           STY  FRETOP
F1A1: 85 70     >1259           STA  FRETOP+1
F1A3: A2 00     >1260           LDX  #0           ;Set program pointer
F1A5: A0 08     >1261           LDY  #8           ; to $800.
F1A7: 86 67     >1262           STX  TXTTAB
F1A9: 84 68     >1263           STY  TXTTAB+1
F1AB: A0 00     >1264           LDY  #0
F1AD: 84 D6     >1265           STY  LOCK         ;Init lock byte and
F1AF: 98        >1266           TYA
F1B0: 91 67     >1267           STA  (TXTTAB),Y ; program beginning byte.
                >1268           >>>  INCR.TXTTAB
F1B2: E6 67     >1268           INC  TXTTAB
F1B4: D0 02     >1268           BNE  NI
F1B6: E6 68     >1268           INC  TXTTAB+1
                >1268 NI        <<<
F1B8: A5 67     >1269           LDA  TXTTAB
F1BA: A4 68     >1270           LDY  TXTTAB+1
F1BC: 20 E3 D3  >1271           JSR  REASON
F1BF: 20 4B D6  >1272           JSR  SCRTCH
                >1273
                >1274 * Now frustrate machine language programmers by
                >1275 * wasting the prime real estate at 0-5:
                >1276
F1C2: A9 3A     >1277           LDA  #STROUT
F1C4: A0 DB     >1278           LDY  #>STROUT
F1C6: 85 04     >1279           STA  GOSTROUT+1 ;Afterthought?
F1C8: 84 05     >1280           STY  GOSTROUT+2
F1CA: A9 3C     >1281           LDA  #RESTART
F1CC: A0 D4     >1282           LDY  #>RESTART
F1CE: 85 01     >1283           STA  GOWARM+1
F1D0: 84 02     >1284           STY  GOWARM+2
F1D2: 6C 01 00  >1285           JMP  (GOWARM+1)
                >1286
                 153            PUT  APSOFT IV
                >1     ********************************
                >2     *                              *
                >3     *    Applesoft - Part IV       *
                >4     *                              *
                >5     * Copywrite Apple Computer,    *
                >6     * Inc. and Microsoft, Inc.;    *
                >7     * not for publication or       *
                >8     * distribution.                *
                >9     *                              *
                >10    ********************************
                >11    *                              *
                >12    *       Graphics, etc.         *
```

```
                        >13   *                              *
                        >14   *        $F1D5 - $F7FF         *
                        >15   *                              *
                        >16   ********************************
                        >17
F1D5: 20 67 DD          >18   CALL      JSR  FRMNUM      ;Note that CALL does not set up
F1D8: 20 52 E7          >19             JSR  GETADR      ; registers as in INTEGER BASIC
F1DB: 6C 50 00          >20             JMP  (LINNUM)    ;On a CALL:
                        >21                              ; Y has low byte CALL adrs
                        >22                              ; A has high byte CALL adrs
                        >23                              ; X has $9D
                        >24
F1DE: 20 F8 E6          >25   INNU      JSR  GETBYT      ;IN#
F1E1: 8A                >26             TXA
F1E2: 4C 8B FE          >27             JMP  INPORT
                        >28
F1E5: 20 F8 E6          >29   PRNU      JSR  GETBYT      ;PR#
F1E8: 8A                >30             TXA
F1E9: 4C 95 FE          >31             JMP  OUTPORT
                        >32
                        >33   * Lores subroutines:
                        >34
                        >35   * Subroutine to get *,* coordinates:
                        >36
F1EC: 20 F8 E6          >37   PLOTFNS   JSR  GETBYT      ;Get first coordinate
F1EF: E0 30             >38             CPX  #48         ;Make sure it is < 48
F1F1: B0 13             >39             BCS  GOERR
F1F3: 86 F0             >40             STX  FIRST
F1F5: A9 2C             >41             LDA  #','
F1F7: 20 C0 DE          >42             JSR  SYNCHR      ;Syntax check
F1FA: 20 F8 E6          >43             JSR  GETBYT      ;Get 2nd coor
F1FD: E0 30             >44             CPX  #48         ;Must be <48
F1FF: B0 05             >45             BCS  GOERR
F201: 86 2C             >46             STX  H2          ;Set up both H2
F203: 86 2D             >47             STX  V2          ; and V2
F205: 60                >48             RTS
                        >49
F206: 4C 99 E1          >50   GOERR     JMP  IQERR
                        >51
F209: 20 EC F1          >52   LINCOOR   JSR  PLOTFNS     ;Get A,B coor
F20C: E4 F0             >53             CPX  FIRST       ;Is B<A?
F20E: B0 08             >54             BCS  AT?         ;Branch if not
F210: A5 F0             >55             LDA  FIRST       ;Interchange them if so
F212: 85 2C             >56             STA  H2
F214: 85 2D             >57             STA  V2
F216: 86 F0             >58             STX  FIRST
F218: A9 C5             >59   AT?       LDA  #at
F21A: 20 C0 DE          >60             JSR  SYNCHR      ;Syntax check
F21D: 20 F8 E6          >61             JSR  GETBYT
F220: E0 30             >62             CPX  #48         ;Check < 48
F222: B0 E2             >63             BCS  GOERR
F224: 60                >64             RTS
                        >65
F225: 20 EC F1          >66   PLOT      JSR  PLOTFNS
F228: 8A                >67             TXA
F229: A4 F0             >68             LDY  FIRST
F22B: C0 28             >69             CPY  #40         ;X coordinate must be <40
```

```
F22D: B0 D7     >70              BCS  GOERR
F22F: 4C 00 F8  >71              JMP  MONPLOT
                >72
F232: 20 09 F2  >73   HLIN       JSR  LINCOOR
  35: 8A        >74              TXA
F236: A4 2C     >75              LDY  H2
F238: C0 28     >76              CPY  #40          ;X coor must be <40
F23A: B0 CA     >77              BCS  GOERR
F23C: A4 F0     >78              LDY  FIRST
F23E: 4C 19 F8  >79              JMP  HLINE        ;Go to monitor routine
                >80
F241: 20 09 F2  >81   VLIN       JSR  LINCOOR
F244: 8A        >82              TXA
F245: A8        >83              TAY
F246: C0 28     >84              CPY  #40          ;X coor must be <40
F248: B0 BC     >85              BCS  GOERR
F24A: A5 F0     >86              LDA  FIRST
F24C: 4C 28 F8  >87              JMP  VLINE        ;Go to monitor routine
                >88
F24F: 20 F8 E6  >89   COLOR      JSR  GETBYT
F252: 8A        >90              TXA
F253: 4C 64 F8  >91              JMP  SETCOL
                >92
                >93   * Non graphics stuff:
                >94
F256: 20 F8 E6  >95   VTAB       JSR  GETBYT
F259: CA        >96              DEX
F25A: 8A        >97              TXA
F25B: C9 18     >98              CMP  #24          ;Range ok?
F25D: B0 A7     >99              BCS  GOERR
F25F: 4C 5B FB  >100             JMP  TABV         ;Use monitor routine
                >101
F262: 20 F8 E6  >102  SPEED      JSR  GETBYT
F265: 8A        >103             TXA               ;$100-SPEED -> SPEEDZ
F266: 49 FF     >104             EOR  #$FF
F268: AA        >105             TAX
F269: E8        >106             INX
F26A: 86 F1     >107             STX  SPEEDZ
F26C: 60        >108             RTS
                >109
F26D: 38        >110  TRACE      SEC
F26E: 90        >111             HEX  90           ;Fake BCC to skip
F26F: 18        >112  NOTRACE    CLC
F270: 66 F2     >113             ROR  TRCFLG
F272: 60        >114             RTS
                >115
F273: A9 FF     >116  NORMAL     LDA  #$FF
F275: D0 02     >117             BNE  NRM
F277: A9 3F     >118  INVERSE    LDA  #$3F
F279: A2 00     >119  NRM        LDX  #0
F27B: 85 32     >120  SI         STA  INVFLG
F27D: 86 F3     >121             STX  ORMASK
F27F: 60        >122             RTS
                >123
F280: A9 7F     >124  FLASH      LDA  #$7F
  82: A2 40     >125             LDX  #$40
F284: D0 F5     >126             BNE  SI
```

```
                    >127
F286: 20 67 DD      >128  HIMEM     JSR  FRMNUM      ;Get specified HIMEM
F289: 20 52 E7      >129            JSR  GETADR
                    >130            >>>  CMPR.LINNUM;STREND
 .8C: A5 50         >130            LDA  LINNUM
F28E: C5 6D         >130            CMP  STREND
F290: A5 51         >130            LDA  LINNUM+1
F292: E5 6E         >130            SBC  STREND+1
                    >130            <<<
F294: B0 03         >131            BGE  SETHI       ;Above variable table?
F296: 4C 10 D4      >132  JMM       JMP  MEMERR      ;Error if not
                    >133
F299: A5 50         >134  SETHI     LDA  LINNUM
F29B: 85 73         >135            STA  MEMSIZ
F29D: 85 6F         >136            STA  FRETOP
F29F: A5 51         >137            LDA  LINNUM+1
F2A1: 85 74         >138            STA  MEMSIZ+1
F2A3: 85 70         >139            STA  FRETOP+1
F2A5: 60            >140            RTS
                    >141
F2A6: 20 67 DD      >142  LOMEM     JSR  FRMNUM
F2A9: 20 52 E7      >143            JSR  GETADR
                    >144            >>>  CMPR.LINNUM;MEMSIZ
F2AC: A5 50         >144            LDA  LINNUM
F2AE: C5 73         >144            CMP  MEMSIZ
F2B0: A5 51         >144            LDA  LINNUM+1
F2B2: E5 74         >144            SBC  MEMSIZ+1
                    >144            <<<
F2B4: B0 E0         >145            BGE  JMM
                    >146            >>>  CMPR.LINNUM;VARTAB
 .2B6: A5 50        >146            LDA  LINNUM
F2B8: C5 69         >146            CMP  VARTAB
F2BA: A5 51         >146            LDA  LINNUM+1
F2BC: E5 6A         >146            SBC  VARTAB+1
                    >146            <<<
F2BE: 90 D6         >147            BLT  JMM
                    >148            >>>  TRDB.LINNUM;VARTAB
                    >148            >>>  TR.LINNUM;VARTAB
F2C0: A5 50         >148            LDA  LINNUM
F2C2: 85 69         >148            STA  VARTAB
                    >148            <<<
                    >148            >>>  TR.LINNUM+1;VARTAB+1
F2C4: A5 51         >148            LDA  LINNUM+1
F2C6: 85 6A         >148            STA  VARTAB+1
                    >148            <<<
                    >148            <<<
F2C8: 4C 6C D6      >149            JMP  CLEARC      ;LOMEM clears variables
                    >150
F2CB: A9 AB         >151  ONERR     LDA  #goto
F2CD: 20 C0 DE      >152            JSR  SYNCHR
                    >153            >>>  TRDB.TXTPTR;TXTPSV
                    >153            >>>  TR.TXTPTR;TXTPSV
F2D0: A5 B8         >153            LDA  TXTPTR
F2D2: 85 F4         >153            STA  TXTPSV
                    >153            <<<
                    >153            >>>  TR.TXTPTR+1;TXTPSV+1
F2D4: A5 B9         >153            LDA  TXTPTR+1
```

```
F2D6: 85 F5     >153            STA  TXTPSV+1
                >153            <<<
                >153            <<<
F2D8: 38        >154            SEC
  D9: 66 D8     >155            ROR  ERRFLG
                >156            >>>  TRDB.CURLIN;CURLSV
                >156            >>>  TR.CURLIN;CURLSV
F2DB: A5 75     >156            LDA  CURLIN
F2DD: 85 F6     >156            STA  CURLSV
                >156            <<<
                >156            >>>  TR.CURLIN+1;CURLSV+1
F2DF: A5 76     >156            LDA  CURLIN+1
F2E1: 85 F7     >156            STA  CURLSV+1
                >156            <<<
                >156            <<<
F2E3: 20 A6 D9  >157            JSR  REMN       ;Ignore rest of line
F2E6: 4C 98 D9  >158            JMP  ADDON
                >159
                >160  * Routine to handle errors if ONERR GOTO active:
                >161
F2E9: 86 DE     >162  HANDLERR  STX  ERRNUM
F2EB: A6 F8     >163            LDX  REMSTK     ;Get stack ptr saved at NEWSTT
F2ED: 86 DF     >164            STX  ERRSTK     ;Remember it
                >165                            ;(Should also have done TXS
                >166                            ; here; see ONERR correction
                >167                            ; in Applesoft manual.)
                >168            >>>  TRDB.CURLIN;ERRLIN
                >168            >>>  TR.CURLIN;ERRLIN
F2EF: A5 75     >168            LDA  CURLIN
  F1: 85 DA     >168            STA  ERRLIN
                >168            <<<
                >168            >>>  TR.CURLIN+1;ERRLIN+1
F2F3: A5 76     >168            LDA  CURLIN+1
F2F5: 85 DB     >168            STA  ERRLIN+1
                >168            <<<
                >168            <<<
                >169            >>>  TRDB.OLDTEXT;ERRPOS
                >169            >>>  TR.OLDTEXT;ERRPOS
F2F7: A5 79     >169            LDA  OLDTEXT
F2F9: 85 DC     >169            STA  ERRPOS
                >169            <<<
                >169            >>>  TR.OLDTEXT+1;ERRPOS+1
F2FB: A5 7A     >169            LDA  OLDTEXT+1
F2FD: 85 DD     >169            STA  ERRPOS+1
                >169            <<<
                >169            <<<
                >170            >>>  TRDB.TXTPSV;TXTPTR
                >170            >>>  TR.TXTPSV;TXTPTR
F2FF: A5 F4     >170            LDA  TXTPSV
F301: 85 B8     >170            STA  TXTPTR
                >170            <<<
                >170            >>>  TR.TXTPSV+1;TXTPTR+1
F303: A5 F5     >170            LDA  TXTPSV+1
F305: 85 B9     >170            STA  TXTPTR+1
                >170            <<<
                >170            <<<
                >171            >>>  TRDB.CURLSV;CURLIN
```

```
                    >171              >>>  TR.CURLSV;CURLIN
F307: A5 F6         >171              LDA  CURLSV
F309: 85 75         >171              STA  CURLIN
                    >171              <<<
                    >171              >>>  TR.CURLSV+1;CURLIN+1
F30B: A5 F7         >171              LDA  CURLSV+1
F30D: 85 76         >171              STA  CURLIN+1
                    >171              <<<
                    >171              <<<
F30F: 20 B7 00      >172              JSR  CHRGOT
F312: 20 3E D9      >173              JSR  GOTO        ;Goto specified ONERR line
F315: 4C D2 D7      >174              JMP  NEWSTT
                    >175
                    >176  RESUME      >>>  TRDB.ERRLIN;CURLIN
                    >176              >>>  TR.ERRLIN;CURLIN
F318: A5 DA         >176              LDA  ERRLIN
F31A: 85 75         >176              STA  CURLIN
                    >176              <<<
                    >176              >>>  TR.ERRLIN+1;CURLIN+1
F31C: A5 DB         >176              LDA  ERRLIN+1
F31E: 85 76         >176              STA  CURLIN+1
                    >176              <<<
                    >176              <<<
                    >177              >>>  TRDB.ERRPOS;TXTPTR
                    >177              >>>  TR.ERRPOS;TXTPTR
F320: A5 DC         >177              LDA  ERRPOS
F322: 85 B8         >177              STA  TXTPTR
                    >177              <<<
                    >177              >>>  TR.ERRPOS+1;TXTPTR+1
F324: A5 DD         >177              LDA  ERRPOS+1
F326: 85 B9         >177              STA  TXTPTR+1
                    >177              <<<
                    >177              <<<
F328: A6 DF         >178              LDX  ERRSTK      ;Retrieve stack ptr as it was
F32A: 9A            >179              TXS              ; when error was encountered
F32B: 4C D2 D7      >180              JMP  NEWSTT
                    >181
F32E: 4C C9 DE      >182  JSYN        JMP  SYNERR
                    >183
F331: B0 FB         >184  DEL         BCS  JSYN        ;Error if # not specified
                    >185              >>>  TRX.PRGEND;VARTAB
F333: A6 AF         >185              LDX  PRGEND
F335: 86 69         >185              STX  VARTAB
F337: A6 B0         >185              LDX  PRGEND+1
F339: 86 6A         >185              STX  VARTAB+1
                    >185              <<<
F33B: 20 0C DA      >186              JSR  LINGET      ;Get beginning of range
F33E: 20 1A D6      >187              JSR  FNDLIN      ;Find this line or next
                    >188              >>>  TRDB.LOWTR;DEST
                    >188              >>>  TR.LOWTR;DEST
F341: A5 9B         >188              LDA  LOWTR
F343: 85 60         >188              STA  DEST
                    >188              <<<
                    >188              >>>  TR.LOWTR+1;DEST+1
F345: A5 9C         >188              LDA  LOWTR+1
F347: 85 61         >188              STA  DEST+1
                    >188              <<<
```

```
                    >188              <<<
F349: A9 2C         >189              LDA  #','
F34B: 20 C0 DE      >190              JSR  SYNCHR      ;Check syntax
F34E: 20 0C DA      >191              JSR  LINGET      ;Get end range
                    >192                               ;(Does nothing if end range
                    >193                               ; not specified.)
                    >194              >>>  INCR.LINNUM ;Point 1 past it
F351: E6 50         >194              INC  LINNUM
F353: D0 02         >194              BNE  NI
F355: E6 51         >194              INC  LINNUM+1
                    >194  NI          <<<
F357: 20 1A D6      >195              JSR  FNDLIN      ;Find next line to it
                    >196              >>>  CMPR.LOWTR;DEST
F35A: A5 9B         >196              LDA  LOWTR
F35C: C5 60         >196              CMP  DEST
F35E: A5 9C         >196              LDA  LOWTR+1
F360: E5 61         >196              SBC  DEST+1
                    >196              <<<
F362: B0 01         >197              BGE  MOVDWN      ;Do move unless
F364: 60            >198              RTS              ;nothing to delete
F365: A0 00         >199  MOVDWN      LDY  #0
F367: B1 9B         >200  MOVIT       LDA  (LOWTR),Y
F369: 91 60         >201              STA  (DEST),Y
                    >202              >>>  INCR.LOWTR
F36B: E6 9B         >202              INC  LOWTR
F36D: D0 02         >202              BNE  NI
F36F: E6 9C         >202              INC  LOWTR+1
                    >202  NI          <<<
                    >203              >>>  INCR.DEST
F371: E6 60         >203              INC  DEST
F373: D0 02         >203              BNE  NI
F375: E6 61         >203              INC  DEST+1
                    >203  NI          <<<
                    >204              >>>  CMPR.VARTAB;LOWTR
F377: A5 69         >204              LDA  VARTAB
F379: C5 9B         >204              CMP  LOWTR
F37B: A5 6A         >204              LDA  VARTAB+1
F37D: E5 9C         >204              SBC  LOWTR+1
                    >204              <<<
F37F: B0 E6         >205              BGE  MOVIT
F381: A6 61         >206              LDX  DEST+1
F383: A4 60         >207              LDY  DEST
F385: D0 01         >208              BNE  NDX
F387: CA            >209              DEX
F388: 88            >210  NDX         DEY
F389: 86 6A         >211              STX  VARTAB+1
F38B: 84 69         >212              STY  VARTAB
F38D: 4C F2 D4      >213              JMP  LINKSET     ;Reset links after a delete
                    >214
F390: AD 56 C0      >215  GR          LDA  LORES
F393: AD 53 C0      >216              LDA  MIXSET
F396: 4C 40 FB      >217              JMP  SETGR
                    >218
F399: AD 54 C0      >219  TEXT        LDA  LOWSCR      ;JMP $FB36 would have
F39C: 4C 39 FB      >220              JMP  SETTXT      ; done both these.
                    >221
                    >222  * Tape array store and recall routines:
```

```
                        >223  * (No corresponding routines for disk in Applesoft.)
                        >224
F39F: 20 D9 F7          >225  STORE     JSR  GETARYPT
F3A2: A0 03             >226            LDY  #3
F3A4: B1 9B             >227            LDA  (LOWTR),Y
F3A6: AA                >228            TAX
F3A7: 88                >229            DEY
F3A8: B1 9B             >230            LDA  (LOWTR),Y
F3AA: E9 01             >231            SBC  #1
F3AC: B0 01             >232            BCS  SLI
F3AE: CA                >233            DEX
F3AF: 85 50             >234  SLI       STA  LINNUM
F3B1: 86 51             >235            STX  LINNUM+1
F3B3: 20 CD FE          >236            JSR  WRITE
F3B6: 20 BC F7          >237            JSR  TAPEPNT
F3B9: 4C CD FE          >238            JMP  WRITE
                        >239
F3BC: 20 D9 F7          >240  RECALL    JSR  GETARYPT
F3BF: 20 FD FE          >241            JSR  MONREAD
F3C2: A0 02             >242            LDY  #2
F3C4: B1 9B             >243            LDA  (LOWTR),Y
F3C6: C5 50             >244            CMP  LINNUM
F3C8: C8                >245            INY
F3C9: B1 9B             >246            LDA  (LOWTR),Y
F3CB: E5 51             >247            SBC  LINNUM+1
F3CD: B0 03             >248            BCS  MOK
F3CF: 4C 10 D4          >249            JMP  MEMERR
F3D2: 20 BC F7          >250  MOK       JSR  TAPEPNT
F3D5: 4C FD FE          >251            JMP  MONREAD
                        >252
                        >253  * Hires initialization routines:
                        >254
F3D8: 2C 55 C0          >255  HGR2      BIT  HISCR
F3DB: 2C 52 C0          >256            BIT  MIXCLR     ;Default to full screen
F3DE: A9 40             >257            LDA  #$40
F3E0: D0 08             >258            BNE  SETHPG
F3E2: A9 20             >259  HGR       LDA  #$20
F3E4: 2C 54 C0          >260            BIT  LOWSCR
F3E7: 2C 53 C0          >261            BIT  MIXSET     ;Default to mixed screen
F3EA: 85 E6             >262  SETHPG    STA  HPAG
F3EC: AD 57 C0          >263            LDA  HIRES
F3EF: AD 50 C0          >264            LDA  TXTCLR
F3F2: A9 00             >265  HCLR      LDA  #0         ;Set for black bkgrnd
F3F4: 85 1C             >266            STA  HCOLOR1
F3F6: A5 E6             >267  BKGND     LDA  HPAG       ;Init hires screen mem
F3F8: 85 1B             >268            STA  SHAPEH
F3FA: A0 00             >269            LDY  #0
F3FC: 84 1A             >270            STY  SHAPEL
F3FE: A5 1C             >271  BKGND1    LDA  HCOLOR1
F400: 91 1A             >272            STA  (SHAPEL),Y ;Clear hires to HCOLOR1
F402: 20 7E F4          >273            JSR  CSHFT2     ;Correct for color shift
F405: C8                >274            INY             ;(Slows clear by factor of 2)
F406: D0 F6             >275            BNE  BKGND1
F408: E6 1B             >276            INC  SHAPEH
F40A: A5 1B             >277            LDA  SHAPEH
F40C: 29 1F             >278            AND  #$1F       ;Done?
F40E: D0 EE             >279            BNE  BKGND1
```

```
F410: 60          >280            RTS
                  >281
                  >282  * Hires position and plot subroutines
                  >283
  11: 85 E2       >284  HPOSN     STA  YO          ;Enter with Y in A-reg
F413: 86 E0       >285            STX  XOL         ; XL in X-reg,
F415: 84 E1       >286            STY  XOH         ; and XH in Y-reg.
F417: 48          >287            PHA
F418: 29 C0       >288            AND  #$C0
F41A: 85 26       >289            STA  GBASL       ;For y-coor = OOABCDEF
F41C: 4A          >290            LSR              ;calculates base addr
F41D: 4A          >291            LSR              ;in GBASL,GBASH for
F41E: 05 26       >292            ORA  GBASL       ;accessing screen memory
F420: 85 26       >293            STA  GBASL       ;via (GBASL),Y.
F422: 68          >294            PLA
F423: 85 27       >295            STA  GBASH
F425: 0A          >296            ASL              ;Calculates
F426: 0A          >297            ASL              ; GBASH = PPPFGHCD,
F427: 0A          >298            ASL              ; GBASL = EABAB000
F428: 26 27       >299            ROL  GBASH       ;where PPP=001 for $2000-$3FFF
F42A: 0A          >300            ASL              ;and PPP=010 for $4000-$5FFF,
F42B: 26 27       >301            ROL  GBASH       ;given y-coor = ABCDEFGH
F42D: 0A          >302            ASL
F42E: 66 26       >303            ROR  GBASL
F430: A5 27       >304            LDA  GBASH
F432: 29 1F       >305            AND  #$1F
F434: 05 E6       >306            ORA  HPAG
F436: 85 27       >307            STA  GBASH
F438: 8A          >308            TXA              ;Divide XO by 7 for
F439: C0 00       >309            CPY  #0          ;index from base adrs (quotient)
F43B: F0 05       >310            BEQ  HPOSN2      ;and bit within screen mem byte
F43D: A0 23       >311            LDY  #$23        ;(mask specified by remainder).
F43F: 69 04       >312            ADC  #4
F441: C8          >313  HPOSN1    INY
F442: E9 07       >314  HPOSN2    SBC  #7
F444: B0 FB       >315            BCS  HPOSN1
F446: 84 E5       >316            STY  HNDX        ;Works for XO from 0 to 279,
F448: AA          >317            TAX              ;low byte in X-reg, high in
F449: BD B9 F4    >318            LDA  MSKTBL-$F9,X
F44C: 85 30       >319            STA  HMASK       ;Y-reg on entry.
F44E: 98          >320            TYA
F44F: 4A          >321            LSR
F450: A5 E4       >322            LDA  HCOLORZ     ;If on odd byte (carry set)
F452: 85 1C       >323            STA  HCOLOR1     ;then rotate bits
F454: B0 28       >324            BCS  CSHFT2
F456: 60          >325            RTS
                  >326
F457: 20 11 F4    >327  HPLOTO    JSR  HPOSN
F45A: A5 1C       >328            LDA  HCOLOR1     ;Calculate bit posn in GBAS,
F45C: 51 26       >329            EOR  (GBASL),Y   ; HNDX, and HMASK from
F45E: 25 30       >330            AND  HMASK       ; Y-coor in A-reg,
F460: 51 26       >331            EOR  (GBASL),Y   ; X-coor in X,Y regs.
F462: 91 26       >332            STA  (GBASL),Y   ;For any 1-bits, substitute
F464: 60          >333            RTS              ;corresp bit of HCOLOR1.
                  >334
                  >335  * Hires L,R,U,D subroutines
                  >336
```

```
F465: 10 23     >337  LFTRT    BPL  RIGHT       ;Use sign for left/right subr.
F467: A5 30     >338  LEFT     LDA  HMASK
F469: 4A        >339           LSR              ;Shift low-order 7 bits
F46A: B0 05     >340           BCS  LEFT1       ; of HMASK one bit posn.
  6C: 49 C0     >341           EOR  #$C0
F46E: 85 30     >342  LR1      STA  HMASK
F470: 60        >343           RTS
F471: 88        >344  LEFT1    DEY              ;Decr horiz index
F472: 10 02     >345           BPL  LEFT2
F474: A0 27     >346           LDY  #$27        ;Wrap around screen
F476: A9 C0     >347  LEFT2    LDA  #$C0        ;New HMASK, rightmost
F478: 85 30     >348  NEWNDX   STA  HMASK       ; dot of byte.
F47A: 84 E5     >349           STY  HNDX        ;Update horiz index.
F47C: A5 1C     >350  CSHIFT   LDA  HCOLOR1
F47E: 0A        >351  CSHFT2   ASL              ;Rotate low-order 7 bits
F47F: C9 C0     >352           CMP  #$C0        ; of HCOLOR1 one bit posn.
F481: 10 06     >353           BPL  RTS1
F483: A5 1C     >354           LDA  HCOLOR1
F485: 49 7F     >355           EOR  #$7F
F487: 85 1C     >356           STA  HCOLOR1
F489: 60        >357  RTS1     RTS
                >358
F48A: A5 30     >359  RIGHT    LDA  HMASK
F48C: 0A        >360           ASL              ;Shift low order 7 bits
F48D: 49 80     >361           EOR  #$80        ; of HMASK one bit posn.
F48F: 30 DD     >362           BMI  LR1
F491: A9 81     >363           LDA  #$81
F493: C8        >364           INY              ;Next byte
F494: C0 28     >365           CPY  #40
 96: 90 E0      >366           BCC  NEWNDX
 98: A0 00      >367           LDY  #0          ;Wrap around
F49A: B0 DC     >368           BCS  NEWNDX      ;Always taken
F49C: 18        >369  LRUDX1   CLC              ;No 90 deg rot
F49D: A5 D1     >370  LRUDX2   LDA  DXH
F49F: 29 04     >371           AND  #4          ;If bit2=0 then no plot
F4A1: F0 25     >372           BEQ  LRUD4
F4A3: A9 7F     >373           LDA  #$7F
F4A5: 25 30     >374           AND  HMASK
F4A7: 31 26     >375           AND  (GBASL),Y   ;Screen bit set?
F4A9: D0 19     >376           BNE  LRUD3
F4AB: E6 EA     >377           INC  COLCOUNT
F4AD: A9 7F     >378           LDA  #$7F
F4AF: 25 30     >379           AND  HMASK
F4B1: 10 11     >380           BPL  LRUD3       ;Always
F4B3: 18        >381  LRUD1    CLC              ;No 90 deg rot
F4B4: A5 D1     >382  LRUD2    LDA  DXH
F4B6: 29 04     >383           AND  #4          ;If bit2=0 then no plot
F4B8: F0 0E     >384           BEQ  LRUD4
F4BA: B1 26     >385           LDA  (GBASL),Y
F4BC: 45 1C     >386           EOR  HCOLOR1     ;Set hires screen bit
F4BE: 25 30     >387           AND  HMASK       ; to corresponding HCOLOR.
F4C0: D0 02     >388           BNE  LRUD3       ;If screen bit changes,
F4C2: E6 EA     >389           INC  COLCOUNT    ;then incr collsn count.
F4C4: 51 26     >390  LRUD3    EOR  (GBASL),Y
F4C6: 91 26     >391           STA  (GBASL),Y
  C8: A5 D1     >392  LRUD4    LDA  DXH         ;Add quadrant to specified
F4CA: 65 D3     >393           ADC  QDRNT       ; vector and move lft, rt,
```

```
F4CC: 29 03     >394             AND  #3          ; up, or down based on
                >395  EQ3        =    *-1         ; sign and carry.
F4CE: C9 02     >396             CMP  #2
F4D0: 6A        >397             ROR
F4D1: B0 92     >398             BCS  LFTRT
F4D3: 30 30     >399  UPDOWN     BMI  DOWN        ;Sign for up/down select.
F4D5: 18        >400  UP         CLC
F4D6: A5 27     >401             LDA  GBASH       ;Calc base adrs for next
F4D8: 2C B9 F5  >402             BIT  EQ1C        ; line up in (GBAS)
F4DB: D0 22     >403             BNE  UP4         ; with 192-line wrap around.
F4DD: 06 26     >404             ASL  GBASL
F4DF: B0 1A     >405             BCS  UP2
F4E1: 2C CD F4  >406             BIT  EQ3
F4E4: F0 05     >407             BEQ  UP1
F4E6: 69 1F     >408             ADC  #$1F        ;**** BIT MAP ****
F4E8: 38        >409             SEC
F4E9: B0 12     >410             BCS  UP3         ;for row = ABCDEFGH
F4EB: 69 23     >411  UP1        ADC  #$23
F4ED: 48        >412             PHA
F4EE: A5 26     >413             LDA  GBASL       ;GBASL = EABAB000
F4F0: 69 B0     >414             ADC  #$B0        ;GBASH = PPPFGHCD
F4F2: B0 02     >415             BCS  UP5
F4F4: 69 F0     >416             ADC  #$F0        ;where PPP=001 for pg1
F4F6: 85 26     >417  UP5        STA  GBASL
F4F8: 68        >418             PLA
F4F9: B0 02     >419             BCS  UP3
F4FB: 69 1F     >420  UP2        ADC  #$1F
F4FD: 66 26     >421  UP3        ROR  GBASL
F4FF: 69 FC     >422  UP4        ADC  #$FC
F501: 85 27     >423  UPDWN1     STA  GBASH
F503: 60        >424             RTS
F504: 18        >425             CLC
F505: A5 27     >426  DOWN       LDA  GBASH
F507: 69 04     >427             ADC  #4          ;Calc base adr for next
                >428  EQ4        =    *-1         ; line down, with 192
F509: 2C B9 F5  >429             BIT  EQ1C        ; line wrap around.
F50C: D0 F3     >430             BNE  UPDWN1
F50E: 06 26     >431             ASL  GBASL
F510: 90 18     >432             BCC  DOWN1
F512: 69 E0     >433             ADC  #$E0
F514: 18        >434             CLC
F515: 2C 08 F5  >435             BIT  EQ4
F518: F0 12     >436             BEQ  DOWN2
F51A: A5 26     >437             LDA  GBASL
F51C: 69 50     >438             ADC  #$50
F51E: 49 F0     >439             EOR  #$F0
F520: F0 02     >440             BEQ  DOWN3
F522: 49 F0     >441             EOR  #$F0
F524: 85 26     >442  DOWN3      STA  GBASL
F526: A5 E6     >443             LDA  HPAG
F528: 90 02     >444             BCC  DOWN2
F52A: 69 E0     >445  DOWN1      ADC  #$E0
F52C: 66 26     >446  DOWN2      ROR  GBASL
F52E: 90 D1     >447             BCC  UPDWN1
                >448
                >449  * Hires line draw subroutines
                >450
```

```
F530: 48        >451  HLINRL   PHA
F531: A9 00     >452          LDA  #0          ;Set XOL,H and YO
F533: 85 E0     >453          STA  XOL         ; to 0 for rel line draw
F535: 85 E1     >454          STA  XOH         ; (DX,DY).
F537: 85 E2     >455          STA  YO
F539: 68        >456          PLA
F53A: 48        >457  HGLIN    PHA             ;On entry:
F53B: 38        >458          SEC              ; XL: A-reg
F53C: E5 E0     >459          SBC  XOL         ; XH: X-reg
F53E: 48        >460          PHA              ; Y:  Y-reg
F53F: 8A        >461          TXA
F540: E5 E1     >462          SBC  XOH
F542: 85 D3     >463          STA  QDRNT       ;Calc ABS(X-XO)
F544: B0 0A     >464          BCS  HLIN2       ; in (DXL,H)
F546: 68        >465          PLA
F547: 49 FF     >466          EOR  #$FF        ;X dir to sign bit
F549: 69 01     >467          ADC  #1          ; of qdrnt.
F54B: 48        >468          PHA              ; 0=right (DX pos)
F54C: A9 00     >469          LDA  #0          ; 1=left  (DX neg)
F54E: E5 D3     >470          SBC  QDRNT
F550: 85 D1     >471  HLIN2    STA  DXH
F552: 85 D5     >472          STA  EH          ;Init EL,H to ABS(X-XO)
F554: 68        >473          PLA
F555: 85 D0     >474          STA  DXL
F557: 85 D4     >475          STA  EL
F559: 68        >476          PLA
F55A: 85 E0     >477          STA  XOL
F55C: 86 E1     >478          STX  XOH
F55E: 98        >479          TYA
F55F: 18        >480          CLC
F560: E5 E2     >481          SBC  YO          ;Calc -ABS(Y-YO)-1 in DY
F562: 90 04     >482          BCC  HLIN3
F564: 49 FF     >483          EOR  #$FF
F566: 69 FE     >484          ADC  #$FE
F568: 85 D2     >485  HLIN3    STA  DY          ;Rotate Y dir into
F56A: 84 E2     >486          STY  YO          ; qdrnt sign bit
F56C: 66 D3     >487          ROR  QDRNT       ; (0=up, 1=down)
F56E: 38        >488          SEC
F56F: E5 D0     >489          SBC  DXL         ;Init COUNTL,H
F571: AA        >490          TAX              ; to -(DELTX,DELTY+1)
F572: A9 FF     >491          LDA  #$FF
F574: E5 D1     >492          SBC  DXH
F576: 85 1D     >493          STA  COUNTH
F578: A4 E5     >494          LDY  HNDX        ;Horiz index.
F57A: B0 05     >495          BCS  MOVEX2      ;Always taken.
F57C: 0A        >496  MOVEX    ASL              ;Move in X-dir.  Use
F57D: 20 65 F4  >497          JSR  LFTRT       ; QDRNT bit6 for lft/rt set.
F580: 38        >498          SEC
F581: A5 D4     >499  MOVEX2   LDA  EL          ;Assume carry set
F583: 65 D2     >500          ADC  DY          ;(EL,H)-DELTY to (EL,H)
F585: 85 D4     >501          STA  EL          ;Note: DY is (-DELTY)-1
F587: A5 D5     >502          LDA  EH          ;Carry clr if (EL,H) goes neg.
F589: E9 00     >503          SBC  #0
F58B: 85 D5     >504  HCOUNT   STA  EH
F58D: B1 26     >505          LDA  (GBASL),Y
F58F: 45 1C     >506          EOR  HCOLOR1     ;Plot dot of HCOLOR1.
F591: 25 30     >507          AND  HMASK
```

```
F593: 51 26     >508            EOR  (GBASL),Y
F595: 91 26     >509            STA  (GBASL),Y
F597: E8        >510            INX                 ;Done (DELTX+DELTY) dots?
F598: D0 04     >511            BNE  HLIN4
  9A: E6 1D     >512            INC  COUNTH
F59C: F0 62     >513            BEQ  RTS2           ;Yes, return.
F59E: A5 D3     >514  HLIN4     LDA  QDRNT          ;For direction test.
F5A0: B0 DA     >515            BCS  MOVEX          ;If carry set, (EL,H) plot
F5A2: 20 D3 F4  >516            JSR  UPDOWN         ; if clr, neg, move
F5A5: 18        >517            CLC
F5A6: A5 D4     >518            LDA  EL             ;(EL,H)+DELTX to (EL,H).
F5A8: 65 D0     >519            ADC  DXL
F5AA: 85 D4     >520            STA  EL
F5AC: A5 D5     >521            LDA  EH
F5AE: 65 D1     >522            ADC  DXH
F5B0: 50 D9     >523            BVC  HCOUNT         ;Always taken
                >524
F5B2: 81        >525  MSKTBL    DFB  %10000001
F5B3: 82        >526            DFB  %10000010
F5B4: 84        >527            DFB  %10000100
F5B5: 88        >528            DFB  %10001000
F5B6: 90        >529            DFB  %10010000
F5B7: A0        >530            DFB  %10100000
F5B8: C0        >531            DFB  %11000000
                >532
F5B9: 1C        >533  EQ1C      HEX  1C
                >534
                >535  * Table of COS(90*X/16 degrees)*$100 - 1 with
                >536  * one byte precision, X=0 to 16:
                >537
F5BA: FF FE FA
F5BD: F4 EC E1
F5C0: D4 C5     >538  COSTBL    HEX  FFFEFAF4ECE1D4C5
F5C2: B4 A1 8D
F5C5: 78 61 49
F5C8: 31 18     >539            HEX  B4A18D7861493118
F5CA: FF        >540            HEX  FF
                >541
                >542  * Hires coordinate restore subroutine:
                >543  * (not called by any Applesoft routine)
                >544
F5CB: A5 26     >545  HFIND     LDA  GBASL          ;Converts base adrs
F5CD: 0A        >546            ASL
F5CE: A5 27     >547            LDA  GBASH          ; to y-coor.
F5D0: 29 03     >548            AND  #3
F5D2: 2A        >549            ROL                 ;For GBASL=EABAB000
F5D3: 05 26     >550            ORA  GBASL          ;    GBASH=PPPFGHCD
F5D5: 0A        >551            ASL
F5D6: 0A        >552            ASL                 ;generate
F5D7: 0A        >553            ASL                 ; Y-coor=ABCDEFGH
F5D8: 85 E2     >554            STA  YO
F5DA: A5 27     >555            LDA  GBASH          ;(PPP=screen page)
F5DC: 4A        >556            LSR
F5DD: 4A        >557            LSR
F5DE: 29 07     >558            AND  #7
  E0: 05 E2     >559            ORA  YO
F5E2: 85 E2     >560            STA  YO             ;Converts HNDX (index from
```

```
F5E4: A5 E5     >561            LDA  HNDX        ; base adrs) and HMASK
F5E6: 0A        >562            ASL              ; (bit mask) to x-coor
F5E7: 65 E5     >563            ADC  HNDX        ; in (XOL,H), range
F5E9: 0A        >564            ASL              ; 0-$133
  EA: AA        >565            TAX
F5EB: CA        >566            DEX
F5EC: A5 30     >567            LDA  HMASK
F5EE: 29 7F     >568            AND  #$7F
F5F0: E8        >569  HFIND1    INX
F5F1: 4A        >570            LSR
F5F2: D0 FC     >571            BNE  HFIND1
F5F4: 85 E1     >572            STA  XOH
F5F6: 8A        >573            TXA
F5F7: 18        >574            CLC              ;Calc HNDX*7+LOG(HMASK), base 2
F5F8: 65 E5     >575            ADC  HNDX
F5FA: 90 02     >576            BCC  HFIND2
F5FC: E6 E1     >577            INC  XOH
F5FE: 85 E0     >578  HFIND2    STA  XOL
F600: 60        >579  RTS2      RTS
                >580
                >581  * Hires shape drawing subroutines
                >582
F601: 86 1A     >583  DRAW0     STX  SHAPEL      ;Shape pointer
F603: 84 1B     >584            STY  SHAPEH
F605: AA        >585  DRAW1     TAX              ;A holds ROT (0-$3F)
F606: 4A        >587            LSR
F607: 4A        >587            LSR
F608: 4A        >587            LSR
F609: 4A        >587            LSR
  0A: 85 D3     >589            STA  QDRNT       ;QDRNT 0=UP, 1=RT
  0C: 8A        >590            TXA              ; 2=DWN, 3=LFT
F60D: 29 0F     >591            AND  #$F
F60F: AA        >592            TAX
F610: BC BA F5  >593            LDY  COSTBL,X    ;Save COS and SIN
F613: 84 D0     >594            STY  DXL         ; values in DXL and DY.
F615: 49 0F     >595            EOR  #$F
F617: AA        >596            TAX
F618: BC BB F5  >597            LDY  COSTBL+1,X
F61B: C8        >598            INY
F61C: 84 D2     >599            STY  DY
F61E: A4 E5     >600            LDY  HNDX        ;Index from base adrs.
F620: A2 00     >601            LDX  #0
F622: 86 EA     >602            STX  COLCOUNT    ;Clear collision count.
F624: A1 1A     >603            LDA  (SHAPEL,X)  ;First byte of shape defn
F626: 85 D1     >604  DRAW2     STA  DXH
F628: A2 80     >605            LDX  #$80
F62A: 86 D4     >606            STX  EL          ;EL,H for fractional
F62C: 86 D5     >607            STX  EH          ; L,R,U,D vectors.
F62E: A6 E7     >608            LDX  SCALEZ      ;Scale factor.
F630: A5 D4     >609  DRAW3     LDA  EL
F632: 38        >610            SEC              ;If frac. cos overflow
F633: 65 D0     >611            ADC  DXL         ; then move in specified
F635: 85 D4     >612            STA  EL          ; vector direction.
F637: 90 04     >613            BCC  DRAW4
F639: 20 B3 F4  >614            JSR  LRUD1
  3C: 18        >615            CLC
F63D: A5 D5     >616  DRAW4     LDA  EH          ;If frac. sin overflow
```

```
F63F: 65 D2     >617            ADC  DY          ; then move in specified
F641: 85 D5     >618            STA  EH          ; direction+90 deg.
F643: 90 03     >619            BCC  DRAW5
F645: 20 B4 F4  >620            JSR  LRUD2
  48: CA        >621  DRAW5     DEX              ;Loop on scale factor.
F649: D0 E5     >622            BNE  DRAW3
F64B: A5 D1     >623            LDA  DXH
F64D: 4A        >624            LSR              ;Next 3 bit vector
F64E: 4A        >625            LSR              ; of shape defn.
F64F: 4A        >626            LSR
F650: D0 D4     >627            BNE  DRAW2       ;Not done this byte.
                >628            >>>  INCR.SHAPEL
F652: E6 1A     >628            INC  SHAPEL
F654: D0 02     >628            BNE  NI
F656: E6 1B     >628            INC  SHAPEL+1
                >628  NI        <<<
F658: A1 1A     >629            LDA  (SHAPEL,X) ;Next byte of shape defn.
F65A: D0 CA     >630            BNE  DRAW2       ;Done if 0.
F65C: 60        >631            RTS
                >632
                >633  * Hires shape xdraw subroutine:
                >634
F65D: 86 1A     >635  XDRAW0    STX  SHAPEL      ;See DRAW comments
F65F: 84 1B     >636            STY  SHAPEH
F661: AA        >637  XDRAW1    TAX
F662: 4A        >639            LSR
F663: 4A        >639            LSR
F664: 4A        >639            LSR
F665: 4A        >639            LSR
  66: 85 D3     >641            STA  QDRNT
  68: 8A        >642            TXA
F669: 29 0F     >643            AND  #$F
F66B: AA        >644            TAX
F66C: BC BA F5  >645            LDY  COSTBL,X
F66F: 84 D0     >646            STY  DXL
F671: 49 0F     >647            EOR  #$F
F673: AA        >648            TAX
F674: BC BB F5  >649            LDY  COSTBL+1,X
F677: C8        >650            INY
F678: 84 D2     >651            STY  DY
F67A: A4 E5     >652            LDY  HNDX
F67C: A2 00     >653            LDX  #0
F67E: 86 EA     >654            STX  COLCOUNT
F680: A1 1A     >655            LDA  (SHAPEL,X)
F682: 85 D1     >656  XDRAW2    STA  DXH
F684: A2 80     >657            LDX  #$80
F686: 86 D4     >658            STX  EL
F688: 86 D5     >659            STX  EH
F68A: A6 E7     >660            LDX  SCALEZ
F68C: A5 D4     >661  XDRAW3    LDA  EL
F68E: 38        >662            SEC
F68F: 65 D0     >663            ADC  DXL
F691: 85 D4     >664            STA  EL
F693: 90 04     >665            BCC  XDRAW4
F695: 20 9C F4  >666            JSR  LRUDX1
  98: 18        >667            CLC
                >668  XDRAW4    >>>  AD.EH;DY;EH
```

```
F699: A5 D5     >668            LDA  EH
F69B: 65 D2     >668            ADC  DY
F69D: 85 D5     >668            STA  EH
                >668            <<<
 9F: 90 03      >669            BCC  XDRAW5
F6A1: 20 9D F4  >670            JSR  LRUDX2
F6A4: CA        >671  XDRAW5    DEX
F6A5: D0 E5     >672            BNE  XDRAW3
F6A7: A5 D1     >673            LDA  DXH
F6A9: 4A        >674            LSR
F6AA: 4A        >675            LSR
F6AB: 4A        >676            LSR
F6AC: D0 D4     >677            BNE  XDRAW2
                >678            >>>  INCR.SHAPEL
F6AE: E6 1A     >678            INC  SHAPEL
F6B0: D0 02     >678            BNE  NI
F6B2: E6 1B     >678            INC  SHAPEL+1
                >678  NI        <<<
F6B4: A1 1A     >679            LDA  (SHAPEL,X)
F6B6: D0 CA     >680            BNE  XDRAW2
F6B8: 60        >681            RTS
                >682
                >683  * Parsing routines used by BASIC for Hires access:
                >684
F6B9: 20 67 DD  >685  HFNS      JSR  FRMNUM
F6BC: 20 52 E7  >686            JSR  GETADR
F6BF: A4 51     >687            LDY  LINNUM+1    ;Get horiz coor in X,Y
F6C1: A6 50     >688            LDX  LINNUM
F6C3: C0 01     >689            CPY  #>280       ;Make sure it is < 280
 C5: 90 06      >690            BLT  HFNS1
 C7: D0 1D      >691            BNE  GGERR
F6C9: E0 18     >692            CPX  #<280
F6CB: B0 19     >693            BGE  GGERR
F6CD: 8A        >694  HFNS1     TXA              ;Save horiz coor on stack
F6CE: 48        >695            PHA
F6CF: 98        >696            TYA
F6D0: 48        >697            PHA
F6D1: A9 2C     >698            LDA  #','
F6D3: 20 C0 DE  >699            JSR  SYNCHR      ;Check syntax
F6D6: 20 F8 E6  >700            JSR  GETBYT      ;Get vert coor
F6D9: E0 C0     >701            CPX  #$C0        ;Check it is < $C0 = 192
F6DB: B0 09     >702            BGE  GGERR
F6DD: 86 9D     >703            STX  DSCTMP      ;Save it
F6DF: 68        >704            PLA              ;Retrieve horiz coor
F6E0: A8        >705            TAY
F6E1: 68        >706            PLA
F6E2: AA        >707            TAX
F6E3: A5 9D     >708            LDA  DSCTMP      ; and vert coor
F6E5: 60        >709            RTS              ;Return to caller
                >710
F6E6: 4C 06 F2  >711  GGERR     JMP  GOERR       ;Illegal quantity
                >712
F6E9: 20 F8 E6  >713  HCOLOR    JSR  GETBYT
F6EC: E0 08     >714            CPX  #8
F6EE: B0 F6     >715            BGE  GGERR
 F0: BD F6 F6   >716            LDA  COLORTBL,X
F6F3: 85 E4     >717            STA  HCOLORZ
```

```
F6F5: 60        >718  RTS3      RTS
              >719
F6F6: 00 2A 55
F6F9: 7F 80 AA
  FC: D5 FF     >720  COLORTBL  HEX   002A557F80AAD5FF
              >721
F6FE: C9 C1     >722  HPLOT     CMP   #to          ;Continued plot requested?
F700: F0 0D     >723            BEQ   HP3          ;Branch if so
F702: 20 B9 F6  >724            JSR   HFNS         ;Get coor of start point
F705: 20 57 F4  >725            JSR   HPLOT0       ;Plot it, setting up coor
F708: 20 B7 00  >726  HP2       JSR   CHRGOT
F70B: C9 C1     >727            CMP   #to          ;Line specified?
F70D: D0 E6     >728            BNE   RTS3         ;Exit if not
F70F: 20 C0 DE  >729  HP3       JSR   SYNCHR
F712: 20 B9 F6  >730            JSR   HFNS         ;Get coor of line end
F715: 84 9D     >731            STY   DSCTMP       ;Set up for line
F717: A8        >732            TAY
F718: 8A        >733            TXA
F719: A6 9D     >734            LDX   DSCTMP
F71B: 20 3A F5  >735            JSR   HGLIN        ;Plot line
F71E: 4C 08 F7  >736            JMP   HP2          ;Loop till no more "TO"
              >737
F721: 20 F8 E6  >738  ROT       JSR   GETBYT
F724: 86 F9     >739            STX   ROTZ
F726: 60        >740            RTS
              >741
F727: 20 F8 E6  >742  SCALE     JSR   GETBYT
F72A: 86 E7     >743            STX   SCALEZ
F72C: 60        >744            RTS
              >745
  2D: 20 F8 E6  >746  DRWPNT    JSR   GETBYT       ;Shape number specified
              >747            >>>   TRDB.SHAPEPNT;SHAPEL
              >747            >>>   TR.SHAPEPNT;SHAPEL
F730: A5 E8     >747            LDA   SHAPEPNT
F732: 85 1A     >747            STA   SHAPEL
              >747            <<<
              >747            >>>   TR.SHAPEPNT+1;SHAPEL+1
F734: A5 E9     >747            LDA   SHAPEPNT+1
F736: 85 1B     >747            STA   SHAPEL+1
              >747            <<<
              >747            <<<
F738: 8A        >748            TXA
F739: A2 00     >749            LDX   #0
F73B: C1 1A     >750            CMP   (SHAPEL,X)   ;Shape defined in table?
F73D: F0 02     >751            BEQ   DP1
F73F: B0 A5     >752            BGE   GGERR        ;Error if not
F741: 0A        >753  DP1       ASL                ;Find address of shape
F742: 90 03     >754            BCC   DP2          ; from table.
F744: E6 1B     >755            INC   SHAPEH
F746: 18        >756            CLC
F747: A8        >757  DP2       TAY
F748: B1 1A     >758            LDA   (SHAPEL),Y
F74A: 65 1A     >759            ADC   SHAPEL
F74C: AA        >760            TAX
F74D: C8        >761            INY
  4E: B1 1A     >762            LDA   (SHAPEL),Y
F750: 65 E9     >763            ADC   SHAPEPNT+1
```

```
F752: 85 1B     >764            STA  SHAPEH      ;Save adrs of shape
F754: 86 1A     >765            STX  SHAPEL
F756: 20 B7 00  >766            JSR  CHRGOT      ;Check syntax
F759: C9 C5     >767            CMP  #at
 5B: D0 09      >768            BNE  DP3
F75D: 20 C0 DE  >769            JSR  SYNCHR
F760: 20 B9 F6  >770            JSR  HFNS        ;Draw it where?
F763: 20 11 F4  >771            JSR  HPOSN
F766: A5 F9     >772  DP3       LDA  ROTZ
F768: 60        >773            RTS
                >774
F769: 20 2D F7  >775  DRAW      JSR  DRWPNT      ;Entries from BASIC
F76C: 4C 05 F6  >776            JMP  DRAW1
F76F: 20 2D F7  >777  XDRAW     JSR  DRWPNT
F772: 4C 61 F6  >778            JMP  XDRAW1
                >779
F775: A9 00     >780  SHLOAD    LDA  #0
F777: 85 3D     >781            STA  A1H
F779: 85 3F     >782            STA  A2H
F77B: A0 50     >783            LDY  #$50
F77D: 84 3C     >784            STY  A1L
F77F: C8        >785            INY
F780: 84 3E     >786            STY  A2L
F782: 20 FD FE  >787            JSR  MONREAD
F785: 18        >788            CLC
F786: A5 73     >789            LDA  MEMSIZ
F788: AA        >790            TAX
F789: CA        >791            DEX
F78A: 86 3E     >792            STX  A2L
 8C: E5 50      >793            SBC  LINNUM
 8E: 48         >794            PHA
F78F: A5 74     >795            LDA  MEMSIZ+1
F791: A8        >796            TAY
F792: E8        >797            INX
F793: D0 01     >798            BNE  SL1
F795: 88        >799            DEY
F796: 84 3F     >800  SL1       STY  A2H
F798: E5 51     >801            SBC  LINNUM+1
F79A: C5 6E     >802            CMP  STREND+1
F79C: 90 02     >803            BCC  SL2
F79E: D0 03     >804            BNE  SL3
F7A0: 4C 10 D4  >805  SL2       JMP  MEMERR
F7A3: 85 74     >806  SL3       STA  MEMSIZ+1
F7A5: 85 70     >807            STA  FRETOP+1
F7A7: 85 3D     >808            STA  A1H
F7A9: 85 E9     >809            STA  SHAPEPNT+1
F7AB: 68        >810            PLA
F7AC: 85 E8     >811            STA  SHAPEPNT
F7AE: 85 73     >812            STA  MEMSIZ
F7B0: 85 6F     >813            STA  FRETOP
F7B2: 85 3C     >814            STA  A1L
F7B4: 20 FA FC  >815            JSR  RD2BIT
F7B7: A9 03     >816            LDA  #3
F7B9: 4C 02 FF  >817            JMP  MONREAD2
                >818
                >819  TAPEPNT   >>>  ADD.LOWTR;LINNUM;A2L
F7BC: 18        >819            CLC
```

```
                        >819            >>>  AD.LOWTR;LINNUM;A2L
F7BD: A5 9B             >819            LDA  LOWTR
F7BF: 65 50             >819            ADC  LINNUM
F7C1: 85 3E             >819            STA  A2L
                        >819            <<<
                        >819            >>>  AD.LOWTR+1;LINNUM+1;A2L+1
F7C3: A5 9C             >819            LDA  LOWTR+1
F7C5: 65 51             >819            ADC  LINNUM+1
F7C7: 85 3F             >819            STA  A2L+1
                        >819            <<<
                        >819            <<<
F7C9: A0 04             >820            LDY  #4
F7CB: B1 9B             >821            LDA  (LOWTR),Y
F7CD: 20 EF E0          >822            JSR  GETARY2
                        >823            >>>  TRDB.HIGHDS;A1L
                        >823            >>>  TR.HIGHDS;A1L
F7D0: A5 94             >823            LDA  HIGHDS
F7D2: 85 3C             >823            STA  A1L
                        >823            <<<
                        >823            >>>  TR.HIGHDS+1;A1L+1
F7D4: A5 95             >823            LDA  HIGHDS+1
F7D6: 85 3D             >823            STA  A1L+1
                        >823            <<<
                        >823            <<<
F7D8: 60                >824            RTS
                        >825
                        >826  * Non graphics stuff:
                        >827
F7D9: A9 40             >828  GETARYPT  LDA  #$40       ;Called by STORE & RECALL
F7DB: 85 14             >829            STA  SUBFLG
F7DD: 20 E3 DF          >830            JSR  PTRGET
F7E0: A9 00             >831            LDA  #0
F7E2: 85 14             >832            STA  SUBFLG
F7E4: 4C F0 D8          >833            JMP  VARTIO
                        >834
                        >835  * Note that if WNDLEFT is not 0, HTAB can pint
                        >836  * outside the screen (eg., in the program).
                        >837
F7E7: 20 F8 E6          >838  HTAB       JSR  GETBYT
F7EA: CA                >839            DEX
F7EB: 8A                >840            TXA
F7EC: C9 28             >841  HTAB1      CMP  #40
F7EE: 90 0A             >842            BLT  HTAB2
F7F0: E9 28             >843            SBC  #40
F7F2: 48                >844            PHA
F7F3: 20 FB DA          >845            JSR  CRDO
F7F6: 68                >846            PLA
F7F7: 4C EC F7          >847            JMP  HTAB1
F7FA: 85 24             >848  HTAB2     STA  CH
F7FC: 60                >849            RTS
                        >850
F7FD: CB D2 D7          >851            HEX  CBD2D7     ;Left over
                        >852


End assembly--
```

10240 bytes

Errors: 0

Symbol table - alphabetical order:

```
    A1H      =$3D        A1L      =$3C        A2H      =$3F        A2L      =$3E
    ABS      =$EBAF   MD AD       =$D000      ADO      =$E7B9      AD1      =$E7C6
    AD2      =$E7CE      AD3      =$E7EA      AD4      =$E7EE      AD5      =$E7FA
MD  ADD      =$D000      ADDACC   =$ECD5      ADDON    =$D998      ADEX2    =$EA10
    ADEX3    =$EA1B      ADEX4    =$EA26      ADEXP    =$EA0E      ADJEXP   =$EC9E
    ADMAN    =$E855      AEX      =$ECA0      AMPER    =$03F5      AND      =$DF55
    ARG      =$A5        ARGSGN   =$AA        ARITH    =$DDC5      ARRAY    =$E11E
    ARYFOUND=$E19E       ARYLOOP  =$E16D      ARYNAM?  =$E179      ARYPNT   =$94
    ARYSTR   =$E50C      ARYTAB   =$6B        ARYVA2   =$E4CA      ARYVA3   =$E4CE
    ARYVAR   =$E4C2      ARYVGO   =$E4D9      ASC      =$E6E5      ASCHEX   =$DA12
    AT?      =$F218      ATN      =$F09E      ATN1     =$F0A6      ATN2     =$F0B4
    ATN3     =$F0C7      ATNSER   =$F0CE      AYINT    =$E10C      AtoFAC   =$E850
    AtoFACS  =$E852      BADNAM   =$DFF4      BADSUBS  =$D2CB      BILLION  =$ED14
    BILmiONE=$ED0F    ?  BKGND    =$F3F6      BKGND1   =$F3FE      BLTU     =$D393
    BLTU2    =$D39A   M  BPD      =$E67F      BREAKIN  =$D35D      BSB      =$E049
MD  BUMP     =$D000      BYPASS   =$E012      CALL     =$F1D5      CANTCON  =$D332
    CAT      =$E597      CH       =$24        CHARAC   =$0D        CHKCLS   =$DEB8
    CHKCOM   =$DEBE      CHKDIM   =$E1AA      CHKDP    =$EC66      CHKMEM   =$D3D6
    CHKNUM   =$DD6A      CHKOPN   =$DEBB      CHKSTR   =$DD6C      CHKTYP   =$DDB4
    CHKVAL   =$DD6D      CHRGET   =$B1        CHRGOT   =$B7        CHRSTR   =$E646
    CLEAR    =$D66A      CLEARC   =$D66C      CMDTABL  =$D000      CMPBM1   =$ED62
    CMPDONE  =$DFC1      CMPHM    =$ED6D      CMPLOOP  =$DFAA   MD CMPR     =$D000
    COLCOUNT=$EA         COLDST   =$F128      COLON?   =$D842      COLOR    =$F24F
    COLORTBL=$F6F6       COMBYTE  =$E74C      COMPARE  =$DDE4      CON      =$D8A1
    CONINT   =$E6FB      CONT     =$D896      CONUPK   =$E9E3      COPSTR   =$DA9A
    COPY     =$DAB7      COS      =$EFEA      COSTBL   =$F5BA      COUNTED  =$EDE7
    COUNTH   =$1D        COUT     =$FDED      CPRMASK  =$16        CPROP    =$DD98
    CPRTYP   =$89        CRDO     =$DAFB      CSHFT2   =$F47E   ?  CSHIFT   =$F47C
    CTRC?    =$D86C      CURLIN   =$75        CURLSV   =$F6        CV2      =$DD74
    CY       =$E45F      DATA     =$D995      DATAFLG  =$13        DATAN    =$D9A3
    DATIN    =$DC69      DATLIN   =$7B        DATPTR   =$7D     MD DECR     =$D000
    DECTBL   =$EE69      DEF      =$E313      DEL      =$F331      DESC?    =$DA8C
    DEST     =$60        DFLTDIM  =$E1F7      DIM      =$DFD9      DIMFLG   =$10
    DIMLUP   =$E253      DIMOK    =$E26F      DIMOK2   =$E270      DIR?     =$DBC7
    DIRCT    =$D7E5      DIV      =$EA5E      DIV10    =$EA55      DIVZ     =$EAE1
    DIVbyZRO=$D2E5       DMTH     =$DE41      DOCMP    =$DFB5      DOERRMSG=$D419
    DOMATH   =$DE43      DOMTH    =$DE3A      DONE     =$D610      DOREENT  =$DB90
    DOSPC    =$DB35      DOWN     =$F505      DOWN1    =$F52A      DOWN2    =$F52C
    DOWN3    =$F524      DP1      =$F741      DP2      =$F747      DP3      =$F766
    DPDIG    =$EC87      DPFLG    =$9B        DPL      =$ED9F      DPLEFT   =$ECA9
    DPLOC    =$ED9E      DPRIGHT  =$ECB2      DRAW     =$F769   ?  DRAWO    =$F601
    DRAW1    =$F605      DRAW2    =$F626      DRAW3    =$F630      DRAW4    =$F63D
    DRAW5    =$F648      DRWPNT   =$F72D      DSCLEN   =$8F        DSCPTR   =$8C
    DSCTMP   =$9D        DV1      =$E538      DV2      =$E542      DVAR     =$E523
    DVARS    =$E519      DVARTS   =$E552      DXH      =$D1        DXL      =$D0
    DY       =$D2        EH       =$D5        EL       =$D4        END      =$D870
    END2     =$D871      END3     =$D888      END4     =$D88A      ENDCHR   =$0E
    ENDFOR   =$DD55      ENDRNG   =$D6C4      EQ1C     =$F5B9      EQ3      =$F4CD
    EQ4      =$F508      EQUAL    =$D0CA      EQUL     =$DED0      EQUOP    =$DE98
    ERFLG?   =$D863      ERLIN    =$DB7F      ERLUP    =$D41F      ERRDIR   =$E306
```

```
ERRFLG   =$D8        ERRIN    =$D350      ERRLIN   =$DA        ERRMSG   =$D260
ERRNUM   =$DE        ERROR    =$D412      ERRPOS   =$DC        ERRSTK   =$DF
EVAL     =$EC61      EVD      =$ECBE      EVDONE   =$ECB9      EXIG     =$DCDF
EXIT     =$DE5D      EXP      =$EF09      EXPON    =$9A        EXPSER   =$EEE0
EXPSGN   =$9C        EXTRAFAC=$AC         EXTRASV  =$92        FAC      =$9D
FACSGN   =$A2        FACSTRNG=$ED36       FADD     =$E7BE      FADDH    =$E7A0
FADDT    =$E7C1      FALSE    =$DF5D      FC1      =$EBE9      FC2      =$EBEF
FCOMP    =$EBB2      FCOMP2   =$EBB4      FD1      =$EA80      FD2      =$EA96
FD3      =$EAA3      FD4      =$EAA6      FD5      =$EAB4      FD6      =$EAD1
FD7      =$EAD5      FDIV     =$EA66      FDIVT    =$EA69      FE1      =$E415
FE2      =$E41F      FEND     =$E3F7      FEVLOOP  =$DD86      FIN      =$EC4A
FIN2     =$EC4E      FIN3     =$EC5D      FINDATA  =$DCA0      FINDEOL  =$D511
FIRST    =$F0        FL1      =$D61E      FL2      =$D635      FLASH    =$F280
FLO1     =$EB9B      FLO2     =$EBA0      FLOAT    =$EB93      FLOOP    =$E832
FM1      =$E9B0      FM2      =$E9B5      FM3      =$E9B8      FM4      =$E9D4
FMU      =$E987      FMULT    =$E97F      FMULTT   =$E982      FN?      =$DEA4
FNC?     =$E341      FNCDATA  =$E3AF      FNCNAM   =$8A        FNDARY   =$E169
FNDELEM  =$E24B      FNDFOR   =$D36A      FNDLIN   =$D61A      FNDMEMHI=$F181
FNDVAR   =$E488      FOR      =$D766      FOR2     =$D777      FORMtoCX=$D31F
FORPNT   =$85        FOUT     =$ED34      FPGEN    =$A4        FPWRT    =$EE97
FR1      =$E874      FR2      =$E880      FR3      =$E88D      FRE      =$E2DE
FRE2     =$E2E5      FREFAC   =$E600      FRESPC   =$71        FRESTR   =$E5FD
FRETMP   =$E604      FRETMS   =$E635      FRETOP   =$6F        FRMEVL   =$DD7B
FRMEVL2  =$DD95      FRMNUM   =$DD67      FROUND   =$E88F      FSUB     =$E7A7
FSUBT    =$E7AA      FULL     =$E474      FUNCT    =$E354      GARBAG   =$E484
GARFLG   =$13        GBASH    =$27        GBASL    =$26        GC       =$D731
GD       =$E0F9      GDBUFS   =$D539      GERR     =$DD0D      GET      =$DBA0
GETADR   =$E752      GETARY   =$E0ED      GETARY2  =$E0EF      GETARYPT=$F7D9
GETBYT   =$E6F8      GETCHR   =$D72C      GETEXP   =$ECE8      GETIN    =$D559
GETLINK  =$D63E      GETLN    =$FD6A      GETOLD   =$E3A9      GETSPA   =$E452
GETSPC   =$E454      GETSTR   =$E6DC      GETVAL   =$DE60      GGERR    =$F6E6
GIQ      =$E948      GIVAYF   =$E2F2      GK       =$D860      GME      =$E26C
GO1      =$D955      GO2      =$D959      GOCMD    =$D828      GOCMD2   =$D82A
GOEND    =$D826      GOERR    =$F206      GOEX     =$DE38      GOGEX    =$EC8A
GOGO     =$E514      GOIQ     =$E6F2      GOLINE   =$D935      GONEWST  =$DD52
GOROUT   =$DF3F      GOSTART  =$D893      GOSTROUT=$03         GOSUB    =$D921
GOTFOR   =$DD0F      GOTO     =$D93E      GOTO?    =$D9F4      GOTSPA   =$E469
GOWARM   =$00        GR       =$F390      GRBPAS   =$E562      GSE      =$E269
GSYNER   =$D981      GTBYTC   =$E6F5      GTFORPNT=$D365       GTLT     =$E011
GTNUM    =$E746      H2       =$2C        HALF     =$EE64      HALFneg  =$E937
HANDLERR=$F2E9   ?   HCLR     =$F3F2      HCOLOR   =$F6E9      HCOLOR1  =$1C
HCOLORZ  =$E4        HCOUNT   =$F58B   ?   HFIND    =$F5CB      HFIND1   =$F5F0
HFIND2   =$F5FE      HFNS     =$F6B9      HFNS1    =$F6CD      HGLIN    =$F53A
HGR      =$F3E2      HGR2     =$F3D8      HIGHDS   =$94        HIGHTR   =$96
HIMEM    =$F286      HIRES    =$C057      HISCR    =$C055      HLIN     =$F232
HLIN2    =$F550      HLIN3    =$F568      HLIN4    =$F59E      HLINE    =$F819
?  HLINRL  =$F530    HMASK    =$30        HMmiTNTH=$ED0A       HNDX     =$E5
HOME     =$FC58      HP2      =$F708      HP3      =$F70F      HPAG     =$E6
HPLOT    =$F6FE      HPLOT0   =$F457      HPOSN    =$F411      HPOSN1   =$F441
HPOSN2   =$F442      HTAB     =$F7E7      HTAB1    =$F7EC      HTAB2    =$F7FA
IF       =$D9C9      IFTRUE   =$D9E1      ILLDIR   =$D2F5      ILLQUAN  =$D295
IN       =$0200      INCHR    =$D553  MD  INCR     =$D000      INDEX    =$5E
INDX     =$99        INLIN    =$D52C      INLIN2   =$D52E      INMSG    =$D358
INNU     =$F1DE      INPDONE  =$DCC6      INPERR   =$DB86      INPFIN   =$DC99
INPORT   =$FE8B      INPRT    =$ED19      INPTR    =$7F        INPUT    =$DBB2
INPUTERR=$DB71       INPUTFLG=$15         INS1     =$E660      INS2     =$E666
INS3     =$E667      INS4     =$E668      INS5     =$E6A2      INSRTDIG=$ECC1
```

```
   INSRTLIN=$D4EA     INSTART =$DC2B     INSTRNG =$E6B9     INT     =$EC23
   INTPART =$ED89     INTVAR? =$E026     INVERSE =$F277     INVFLG  =$32
   IOERR   =$E199     ISCNTC  =$D858     ISLETC  =$E07D     ISTOK?  =$D590
   JARY    =$E046     JD10    =$ED7F     JER     =$E19B     JERR    =$E432
   JERROR  =$DD78     JF      =$DFCA     JGOCMD  =$D9E9     JLNK    =$D8ED
   JM10    =$ED78     JMM     =$F296     JMPADRS =$90       JOV     =$EA36
   JSY     =$D846     JSYN    =$F32E     KEY     =$C000     LASTOP  =$87
   LASTPT  =$53    ?  LEFT    =$F467     LEFT1   =$F471     LEFT2   =$F476
   LEFTSTR =$E65A     LEN     =$E6D6     LENGTH  =$91       LET     =$DA46
   LET2    =$DA63     LETREAL =$DA77     LETSTR  =$DA7A     LFTRT   =$F465
   LG2     =$E94B     LIN     =$D5A8     LINCOOR =$F209     LINGET  =$DA0C
   LINKSET =$D4F2     LINNUM  =$50       LINPRT  =$ED24     LIST    =$D6A5
   LISTED  =$D724     LISTLOOP=$D6FE     LOAD    =$D8C9     LOCK    =$D6
   LOG     =$E941     LOGSER  =$E918     LOGe    =$EEDB     LOGtwo  =$E93C
   LOMEM   =$F2A6     LORES   =$C056     LOWSCR  =$C054     LOWTR   =$9B
   LR1     =$F46E     LRUD1   =$F4B3     LRUD2   =$F4B4     LRUD3   =$F4C4
   LRUD4   =$F4C8     LRUDX1  =$F49C     LRUDX2  =$F49D     LST1LIN =$D6F7
   LSTD?   =$D6F5     MAF     =$EB66     MAF2    =$EB68     MAININP =$DBEB
   MAINLST =$D6CC     MAKARY  =$E1C1     MAKDIGIT=$EDEE     MAKINT  =$E102
   MAKSTR  =$EDBD     MARKEND =$EE5A     MATHTBL =$D0B2     MB      =$ED57
   MEMERR  =$D410     MEMFOUND=$F195     MEMSIZ  =$73       MFA     =$EB55
   MFA2    =$EB59     MI1     =$E119     MI2     =$E11B     MIDSTR  =$E691
   MIN     =$DECE     MINUS   =$D0C7     MISMTCH =$DD76     MIXCLR  =$C052
   MIXSET  =$C053     MKINT   =$E108     MOK     =$F3D2     MONPLOT =$F800
   MONREAD =$FEFD     MONREAD2=$FF02     MONWAIT =$FCA8     MORNAM  =$E007
   MOV1F   =$EB21     MOV2F   =$EB1E     MOVAF   =$EB63     MOVDWN  =$F365
   MOVESTR =$E5E6     MOVEX   =$F57C     MOVEX2  =$F581     MOVFA   =$EB53
   MOVFM   =$EAF9     MOVINS  =$E5D4     MOVIT   =$F367     MOVMF   =$EB2B
?  MOVML   =$EB23     MOVSTR  =$E5E2     MSKTBL  =$F5B2     MSLUP   =$EDC1
   MU1     =$E2B6     MU2     =$E2C0     MU3     =$E2D9     MUL10   =$EA39
   MULT    =$E2AD     MVBACK  =$EE19     MVBYT   =$D3C3     MVDG    =$ECF7
   MVDWN   =$D4A7     MVPRG   =$D4D1     MVS2    =$E5EA     MVS3    =$E5F3
   MVZP    =$F152     NAMOK   =$DFF7     NB      =$E630     NCR     =$D712
M  ND      =$DDEE     NDP     =$ECC8     NDX     =$F388     NEEDEX? =$EE26
MD NEG     =$D000     NEG2    =$E8A4     NEGATE  =$DB00     NEGFAC  =$E89E
   NEGNUM  =$E0FE     NEGOP   =$EED0     NEW     =$D649     NEWDESC =$DAA1
   NEWLN?  =$D4B5     NEWNDX  =$F478     NEWSTT  =$D7D2     NEWVAR  =$E09C
   NEXT    =$DCF9     NEXTCHR =$DB2F  M  NI      =$F6B4     NIN     =$E036
   NINT    =$E1D5     NINTA   =$E292     NL1     =$D49F     NOI     =$D54C
   NORMAL  =$F273     NOSUCH  =$D647     NOT?    =$DE90     NOTFND  =$E087
   NOTFOUND=$E1B8     NOTMATH =$DE35     NOTOK   =$D83F     NOTRACE =$F26F
   NOTZ    =$DE9F     NOTZE   =$ED4F     NOTinstr=$DF3A     NRM     =$F279
   NTD     =$DCD1     NTL     =$E5B7     NUM10   =$EA50     NUMBER  =$DE69
   NUMCMP  =$DFB0     NUMDIM  =$0F       NUMIN   =$DC63     NV1     =$E05B
   NV2     =$E065     NWV     =$E0B2     NX      =$D5A7     NXARY   =$E188
   NXBYT   =$D3C7     NXCHAR  =$DB44     NXCHR   =$D56D     NXCLM   =$DB0E
   NXDIG   =$DA40     NXDIGIT =$EC64     NXDIM   =$DFD6     NXDM    =$E281
   NXFOR   =$D38B     NXIN    =$DBDC     NXINP   =$DBF1     NXLIN   =$D45C
   NXLINK  =$D4FE     NXLST   =$D6DA     NXNUM   =$DA00     NXOP    =$DDD6
   NXPAG   =$D3CE     NXPG    =$E22E     NXPTR   =$E073     NXS     =$DCB9
   NXSFT   =$E8DC     NXSPC   =$DB2C     NXTDIM  =$E12C     NXTERM  =$EF96
   NXVAR   =$E059     NXwoFOR =$D260     NY      =$D5A2     NZ      =$E409
   ODDSER  =$EF5C     OLDLIN  =$77       OLDTEXT =$79       ONCNT   =$D9F8
   ONE     =$E913     ONERR   =$F2CB     ONESTEP =$D7C3     ONGOTO  =$D9EC
   OOR     =$EF24     OR      =$DF4F     ORMASK  =$F3       OUTDO   =$DB5C
   OUTOFRNG=$EA2B     OUTPORT =$FE95     OUTQUES =$DB5A     OUTSP   =$DB57
   OVERFLOW=$E8D5     OVFLOW  =$D2A5     OofDATA =$D28A     OofMEM  =$D2AD
```

```
PARCHK   =$DEB2      PARITY?  =$EDE5      PARSE    =$D56C      PDL      =$DFCD
PECHR    =$DC4C      PEEK     =$E764      PENCHR   =$DC4B      PIdoub   =$F06B
PIhalf   =$F066      PLOT     =$F225      PLOTFNS  =$F1EC      PLU?     =$D604
PLUS     =$D0CD      PLUSEPS  =$E8C6      PNTR     =$0F        PNTSTK   =$EE5F
POINT    =$E73D      POKE     =$E77B      POP      =$D96B      POS      =$E2FF
POSOP    =$DF65      PREAD    =$FB1E      PREFNC   =$DDF6      PREFTEST=$DDCD
PRGEND   =$AF        PRINT    =$DAD5      PRINT2   =$DAD7    ?  PRNTFAC  =$ED2E
PRNTIN?  =$D431      PRNU     =$F1E5      PROGIO   =$D901      PROMPT   =$33
PRSTR    =$ED31      PRSTRING=$DACF      PRTOK    =$D750      PSHF     =$DE10
PSHFACX  =$DE15      PSHMAD   =$DDFD      PTRGET   =$DFE3      PTRGET2  =$DFE8
PTRGET3  =$DFEA   MD PULL     =$D000      PULL3    =$D9C5   MD PUSH     =$D000
PUSHFAC  =$DE20      PUTCHR   =$DC3F      PUTDP    =$EDAA      PUTEMP   =$E435
PUTEX    =$EE36      PUTIN    =$D5CD      PUTLINK  =$D50F      PUTNEW   =$E42A
PUTSTR   =$DA7B      PUTTOK   =$D5CB      PW1      =$EEA0      PW2      =$EEBA
QDRNT    =$D3        QI1      =$EC06      QI2      =$EC12      QINT     =$EBF2
QOUT     =$DBC4      QUARTER  =$F070      QUO?     =$E404      RAR      =$E1DE
RARY     =$E298      RD1      =$EFCC      RD2      =$EFE7      RD2BIT   =$FCFA
RDKEY    =$FD0C      READ     =$DBE2      READERR  =$DB7B      REASON   =$D3E3
RECALL   =$F3BC      REENT    =$DCEF      REM      =$D9DC      REM?     =$D5E2
REMN     =$D9A6      REMSTK   =$F8        RES>FAC  =$EAE6      RESPERR  =$DB87
RESTART  =$D43C      RESTORE  =$D849      RESULT   =$62        RESUME   =$F318
RET1     =$D392      RET10    =$DB02      RET11    =$DCDE      RET12    =$DD73
RET2     =$D40F      RET3     =$D648      RET4     =$D696      RET5     =$D857
RET6     =$D8AF      RET7     =$D96A      RET8     =$D9A2      RET9     =$DA0B
RETURN   =$D984      REdimARR=$D2D8      RIGHT    =$F48A      RIGHTSTR=$E686
RM1      =$D9AE      RM2      =$D9B6      RND      =$EFAE      RNDADJ1  =$EFA6
RNDADJ2  =$EFAA      RNDB     =$EB72      RNDSEED  =$C9        ROT      =$F721
ROTZ     =$F9        ROUN     =$ED86      ROUND    =$EB7A      RS1      =$D3ED
RS2      =$D3F1      RS3      =$D3FC      RTN1     =$E086      RTN10    =$EB8F
RTN11    =$EC11      RTN12    =$EC49      RTN13    =$EEDA      RTN14    =$EFA5
RTN15    =$F0CD      RTN16    =$F122      RTN2     =$E2AC      RTN3     =$E645
RTN4     =$E79F      RTN5     =$E89D      RTN6     =$E8D4      RTN7     =$E9E2
RTN8     =$EA4F      RTN9     =$EB71      RTNwoGSB=$D276       RTS1     =$F489
RTS2     =$F600      RTS3     =$F6F5      RUN      =$D912      RUNLINE  =$D91B
SAMEFOR?=$D37F       SAVDIM   =$E1E7      SAVE     =$D8B0      SAVEOBJ  =$00
SAVOLD   =$E378      SAVOP    =$DDD7      SAVY     =$EE09   MD SB       =$D000
SCALE    =$F727      SCALEZ   =$E7        SCDCH    =$E03D      SCREEN   =$DEF9
SCRN     =$F871      SCRTCH   =$D64B      SE       =$D578      SEND     =$DB64
SENDCHR  =$D702      SERIES   =$EF72      SERLEN   =$A3        SERLOOP  =$EF89
SERMAIN  =$EF76      SERPNT   =$AD        SETCOL   =$F864      SETDA    =$D853
SETDP    =$EC98      SETEND   =$D3B7      SETFOR   =$EB27      SETGR    =$FB40
SETHI    =$F299      SETHPG   =$F3EA      SETPTRS  =$D665      SETSGN   =$EC85
SETTXT   =$FB39      SETVPNT  =$E0DE      SFS      =$DFA5      SFSG     =$ED41
SGN      =$EB90      SGN?     =$DEAB      SGNCHK   =$EC8C      SGNCPR   =$AB
SGNIF    =$E829      SH1      =$E8FD      SH2      =$E903      SH3      =$E911
SHAPEH   =$1B        SHAPEL   =$1A        SHAPEPNT=$E8         SHFTIN   =$D5E9
SHFTR    =$E907      SHFTRES  =$E8DA      SHIFT    =$E8F0      SHIN     =$D5F2
SHLOAD   =$F775      SI       =$F27B      SI1      =$F023      SI2      =$F026
SI3      =$F033      SIGN     =$EB82      SIGN1    =$EB86      SIGN2    =$EB88
SIGNFLG  =$16        SIGNIF   =$E82E      SIN      =$EFF1      SINSER   =$F075
SK2      =$D5FD      SKIP     =$DE64      SKIPTOK  =$D5F9      SKP      =$DC57
SKPTK    =$D746      SKPV     =$DD02      SL1      =$F796      SL2      =$F7A0
SL3      =$F7A3      SLI      =$F3AF      SNDQ?    =$DC1F      SNGFLT   =$E301
SNTXERR  =$DE0D      SPC?     =$DB21      SPEED    =$F262      SPEEDZ   =$F1
SQR      =$EE8D      SQRhalf  =$E92D      SQRtwo   =$E932      SS       =$EF85
SSF      =$D5E0      ST1      =$DE8A      STACK    =$0100      STE      =$ED60
STEP     =$D7AF      STEX     =$ED05      STKINI   =$D683      STOP     =$D86E
```

```
   STORE    =$F39F      STR      =$E3C5      STRCMP   =$DF7D      STREND   =$6D
   STRINI   =$E3D5      STRIP    =$D541      STRLIT   =$E3E7      STRLT2   =$E3ED
   STRNG1   =$AB        STRNG2   =$AD        STRNG?   =$E01C      STROUT   =$DB3A
   STRPRT   =$DB3D      STRSPA   =$E3DD      STRTRNG  =$D6B1      STRTXT   =$DE81
   STRtoLNG=$D310       STXP     =$DC27      STXTPT   =$D697   MD SUB      =$D000
   SUBERR   =$E196      SUBFLG   =$14        SUBMAN   =$E806      SVAR     =$E4B5
   SVARGO   =$E4BD      SVARS    =$E4A9      SVP      =$E0E8      SVY      =$EDBB
   SWPNT    =$DC7E      SYNCHR   =$DEC0      SYNERR   =$DEC9      SYNTXERR=$D270
   TAB      =$DB03      TABIT    =$DB2B      TABV     =$FB5B      TABWHERE=$DB16
   TAN      =$F03A      TAN2     =$F062      TAPEPNT  =$F7BC      TEMP1    =$93
   TEMP2    =$98        TEMP3    =$8A        TEMPPT   =$52        TEMPST   =$55
   TEND     =$EE8D      TEXT     =$F399      TMPEXP   =$99        TOK?     =$D588
   TOKDONE  =$D75F      TOKEN?   =$D734      TOKLP    =$D749      TOKLUP   =$D755
   TOKTABL  =$D0D0   MD TR       =$D000      TR1      =$D81D      TRACE    =$F26D
   TRACE?   =$D805   MD TRAX     =$D000   MD TRAY     =$D000      TRCFLG   =$F2
MD TRDB     =$D000      TRUE     =$DF60      TRUE?    =$D9D8   MD TRX      =$D000
MD TRXY     =$D000      TVAR     =$E4A0      TWOBRK   =$E09A      TXPSV    =$87
   TXTCLR   =$C050      TXTPSV   =$F4        TXTPTR   =$B8        TXTTAB   =$67
   TYPEMISS=$D303       UNARY    =$DF0C      UNDERR   =$D97C      UNDFNC   =$E30E
   UNDFUNC  =$D340      UNDSTAT  =$D2BA      UNFNC    =$D080   ?  UP       =$F4D5
   UP1      =$F4EB      UP2      =$F4FB      UP3      =$F4FD      UP4      =$F4FF
   UP5      =$F4F6      UPDOWN   =$F4D3      UPDWN1   =$F501      USR      =$0A
   V2       =$2D        VAL      =$E707      VALTYP   =$11        VAR?     =$DE6C
   VARL     =$DED5      VARNAM   =$81        VARNXT   =$DCFF      VARPNT   =$83
   VARTAB   =$69        VARTIO   =$D8F0      VDONE    =$E55D      VL2      =$E70F
   VL3      =$E727      VLIN     =$F241      VLINE    =$F828      VPNT     =$A0
   VR1      =$DEE5      VR2      =$DEF6      VSEARCH  =$E04F      VTAB     =$F256
   WAIT     =$E784      WHATX    =$EE42      WNDUP    =$EE57      WNX      =$DC72
   WRITE    =$FECD      WT2      =$E793      WT3      =$E797      X0H      =$E1
   X0L      =$E0        X1       =$EF19      X2       =$EF27      X3       =$EF37
   XDRAW    =$F76F   ?  XDRAW0   =$F65D      XDRAW1   =$F661      XDRAW2   =$F682
   XDRAW3   =$F68C      XDRAW4   =$F699      XDRAW5   =$F6A4      Y0       =$E2
   ZARY     =$E21A      ZEND     =$E408      ZERO     =$EA31      ZEROFAC  =$E84E
   ZF       =$DBE9      ZFAC     =$EC40      ZLUP     =$E229      ZPSTUFF  =$F10B
   at       =$C5        data     =$83        equal    =$D0        fn       =$C2
   for      =$81        gosub    =$B0        goto     =$AB        leftstr  =$E8
   minus    =$C9        not      =$C6        plus     =$C8        pop      =$A1
   print    =$BA        rem      =$B2        scrn     =$D7        sgn      =$D2
   spc      =$C3        step     =$C7        tab      =$C0        then     =$C4
   to       =$C1
```

Symbol table - numerical order:

```
   SAVEOBJ =$00        GOWARM   =$00        GOSTROUT=$03        USR      =$0A
   CHARAC  =$0D        ENDCHR   =$0E        PNTR     =$0F        NUMDIM   =$0F
   DIMFLG  =$10        VALTYP   =$11        DATAFLG  =$13        GARFLG   =$13
   SUBFLG  =$14        INPUTFLG=$15         CPRMASK  =$16        SIGNFLG  =$16
   SHAPEL  =$1A        SHAPEH   =$1B        HCOLOR1  =$1C        COUNTH   =$1D
   CH      =$24        GBASL    =$26        GBASH    =$27        H2       =$2C
   V2      =$2D        HMASK    =$30        INVFLG   =$32        PROMPT   =$33
   A1L     =$3C        A1H      =$3D        A2L      =$3E        A2H      =$3F
   LINNUM  =$50        TEMPPT   =$52        LASTPT   =$53        TEMPST   =$55
   INDEX   =$5E        DEST     =$60        RESULT   =$62        TXTTAB   =$67
   VARTAB  =$69        ARYTAB   =$6B        STREND   =$6D        FRETOP   =$6F
   FRESPC  =$71        MEMSIZ   =$73        CURLIN   =$75        OLDLIN   =$77
   OLDTEXT =$79        DATLIN   =$7B        DATPTR   =$7D        INPTR    =$7F
```

```
   for       =$81        VARNAM   =$81        data      =$83        VARPNT   =$83
   FORPNT    =$85        TXPSV    =$87        LASTOP    =$87        CPRTYP   =$89
   FNCNAM    =$8A        TEMP3    =$8A        DSCPTR    =$8C        DSCLEN   =$8F
   JMPADRS   =$90        LENGTH   =$91        EXTRASV   =$92        TEMP1    =$93
   ARYPNT    =$94        HIGHDS   =$94        HIGHTR    =$96        TEMP2    =$98
   INDX      =$99        TMPEXP   =$99        EXPON     =$9A        LOWTR    =$9B
   DPFLG     =$9B        EXPSGN   =$9C        DSCTMP    =$9D        FAC      =$9D
   VPNT      =$A0        pop      =$A1        FACSGN    =$A2        SERLEN   =$A3
   FPGEN     =$A4        ARG      =$A5        ARGSGN    =$AA        goto     =$AB
   SGNCPR    =$AB        STRNG1   =$AB        EXTRAFAC=$AC         SERPNT   =$AD
   STRNG2    =$AD        PRGEND   =$AF        gosub     =$B0        CHRGET   =$B1
   rem       =$B2        CHRGOT   =$B7        TXTPTR    =$B8        print    =$BA
   tab       =$C0        to       =$C1        fn        =$C2        spc      =$C3
   then      =$C4        at       =$C5        not       =$C6        step     =$C7
   plus      =$C8        minus    =$C9        RNDSEED   =$C9        equal    =$D0
   DXL       =$D0        DXH      =$D1        sgn       =$D2        DY       =$D2
   QDRNT     =$D3        EL       =$D4        EH        =$D5        LOCK     =$D6
   scrn      =$D7        ERRFLG   =$D8        ERRLIN    =$DA        ERRPOS   =$DC
   ERRNUM    =$DE        ERRSTK   =$DF        XOL       =$E0        XOH      =$E1
   YO        =$E2        HCOLORZ  =$E4        HNDX      =$E5        HPAG     =$E6
   SCALEZ    =$E7        leftstr  =$E8        SHAPEPNT=$E8         COLCOUNT=$EA
   FIRST     =$F0        SPEEDZ   =$F1        TRCFLG    =$F2        ORMASK   =$F3
   TXTPSV    =$F4        CURLSV   =$F6        REMSTK    =$F8        ROTZ     =$F9
   STACK     =$0100      IN       =$0200      AMPER     =$03F5      KEY      =$C000
   TXTCLR    =$C050      MIXCLR   =$C052      MIXSET    =$C053      LOWSCR   =$C054
   HISCR     =$C055      LORES    =$C056      HIRES     =$C057   MD INCR     =$D000
MD DECR      =$D000   MD CMPR      =$D000   MD TR        =$D000   MD TRDB     =$D000
MD TRX       =$D000   MD TRAX      =$D000   MD TRAY      =$D000   MD TRXY     =$D000
MD AD        =$D000   MD ADD       =$D000   MD SB        =$D000   MD SUB      =$D000
   BUMP      =$D000   MD PUSH      =$D000   MD PULL      =$D000   MD NEG      =$D000
   CMDTABL   =$D000      UNFNC    =$D080      MATHTBL   =$D0B2      MINUS    =$D0C7
   EQUAL     =$D0CA      PLUS     =$D0CD      TOKTABL   =$D0D0      ERRMSG   =$D260
   NXwoFOR   =$D260      SYNTXERR=$D270       RTNwoGSB=$D276        OofDATA  =$D28A
   ILLQUAN   =$D295      OVFLOW   =$D2A5      OofMEM    =$D2AD      UNDSTAT  =$D2BA
   BADSUBS   =$D2CB      REdimARR=$D2D8       DIVbyZRO=$D2E5        ILLDIR   =$D2F5
   TYPEMISS=$D303        STRtoLNG=$D310       FORMtoCX=$D31F        CANTCON  =$D332
   UNDFUNC   =$D340      ERRIN    =$D350      INMSG     =$D358      BREAKIN  =$D35D
   GTFORPNT=$D365        FNDFOR   =$D36A      SAMEFOR?=$D37F        NXFOR    =$D38B
   RET1      =$D392      BLTU     =$D393      BLTU2     =$D39A      SETEND   =$D3B7
   MVBYT     =$D3C3      NXBYT    =$D3C7      NXPAG     =$D3CE      CHKMEM   =$D3D6
   REASON    =$D3E3      RS1      =$D3ED      RS2       =$D3F1      RS3      =$D3FC
   RET2      =$D40F      MEMERR   =$D410      ERROR     =$D412      DOERRMSG=$D419
   ERLUP     =$D41F      PRNTIN?  =$D431      RESTART   =$D43C      NXLIN    =$D45C
   NL1       =$D49F      MVDWN    =$D4A7      NEWLN?    =$D4B5      MVPRG    =$D4D1
   INSRTLIN=$D4EA        LINKSET  =$D4F2      NXLINK    =$D4FE      PUTLINK  =$D50F
   FINDEOL   =$D511      INLIN    =$D52C      INLIN2    =$D52E      GDBUFS   =$D539
   STRIP     =$D541      NOI      =$D54C      INCHR     =$D553      GETIN    =$D559
   PARSE     =$D56C      NXCHR    =$D56D      SE        =$D578      TOK?     =$D588
   ISTOK?    =$D590      NY       =$D5A2      NX        =$D5A7      LIN      =$D5A8
   PUTTOK    =$D5CB      PUTIN    =$D5CD      SSF       =$D5E0      REM?     =$D5E2
   SHFTIN    =$D5E9      SHIN     =$D5F2      SKIPTOK   =$D5F9      SK2      =$D5FD
   PLU?      =$D604      DONE     =$D610      FNDLIN    =$D61A      FL1      =$D61E
   FL2       =$D635      GETLINK  =$D63E      NOSUCH    =$D647      RET3     =$D648
   NEW       =$D649      SCRTCH   =$D64B      SETPTRS   =$D665      CLEAR    =$D66A
   CLEARC    =$D66C      STKINI   =$D683      RET4      =$D696      STXTPT   =$D697
   LIST      =$D6A5      STRTRNG  =$D6B1      ENDRNG    =$D6C4      MAINLST  =$D6CC
   NXLST     =$D6DA      LSTD?    =$D6F5      LST1LIN   =$D6F7      LISTLOOP=$D6FE
```

```
SENDCHR =$D702   NCR     =$D712   LISTED  =$D724   GETCHR  =$D72C
GC      =$D731   TOKEN?  =$D734   SKPTK   =$D746   TOKLP   =$D749
PRTOK   =$D750   TOKLUP  =$D755   TOKDONE =$D75F   FOR     =$D766
FOR2    =$D777   STEP    =$D7AF   ONESTEP =$D7C3   NEWSTT  =$D7D2
DIRCT   =$D7E5   TRACE?  =$D805   TR1     =$D81D   GOEND   =$D826
GOCMD   =$D828   GOCMD2  =$D82A   NOTOK   =$D83F   COLON?  =$D842
JSY     =$D846   RESTORE =$D849   SETDA   =$D853   RET5    =$D857
ISCNTC  =$D858   GK      =$D860   ERFLG?  =$D863   CTRC?   =$D86C
STOP    =$D86E   END     =$D870   END2    =$D871   END3    =$D888
END4    =$D88A   GOSTART =$D893   CONT    =$D896   CON     =$D8A1
RET6    =$D8AF   SAVE    =$D8B0   LOAD    =$D8C9   JLNK    =$D8ED
VARTIO  =$D8F0   PROGIO  =$D901   RUN     =$D912   RUNLINE =$D91B
GOSUB   =$D921   GOLINE  =$D935   GOTO    =$D93E   GO1     =$D955
GO2     =$D959   RET7    =$D96A   POP     =$D96B   UNDERR  =$D97C
GSYNER  =$D981   RETURN  =$D984   DATA    =$D995   ADDON   =$D998
RET8    =$D9A2   DATAN   =$D9A3   REMN    =$D9A6   RM1     =$D9AE
RM2     =$D9B6   PULL3   =$D9C5   IF      =$D9C9   TRUE?   =$D9D8
REM     =$D9DC   IFTRUE  =$D9E1   JGOCMD  =$D9E9   ONGOTO  =$D9EC
GOTO?   =$D9F4   ONCNT   =$D9F8   NXNUM   =$DA00   RET9    =$DA0B
LINGET  =$DA0C   ASCHEX  =$DA12   NXDIG   =$DA40   LET     =$DA46
LET2    =$DA63   LETREAL =$DA77   LETSTR  =$DA7A   PUTSTR  =$DA7B
DESC?   =$DA8C   COPSTR  =$DA9A   NEWDESC =$DAA1   COPY    =$DAB7
PRSTRING=$DACF   PRINT   =$DAD5   PRINT2  =$DAD7   CRDO    =$DAFB
NEGATE  =$DB00   RET10   =$DB02   TAB     =$DB03   NXCLM   =$DB0E
TABWHERE=$DB16   SPC?    =$DB21   TABIT   =$DB2B   NXSPC   =$DB2C
NEXTCHR =$DB2F   DOSPC   =$DB35   STROUT  =$DB3A   STRPRT  =$DB3D
NXCHAR  =$DB44   OUTSP   =$DB57   OUTQUES =$DB5A   OUTDO   =$DB5C
SEND    =$DB64   INPUTERR=$DB71   READERR =$DB7B   ERLIN   =$DB7F
INPERR  =$DB86   RESPERR =$DB87   DOREENT =$DB90   GET     =$DBA0
INPUT   =$DBB2   QOUT    =$DBC4   DIR?    =$DBC7   NXIN    =$DBDC
READ    =$DBE2   ZF      =$DBE9   MAININP  =$DBEB   NXINP   =$DBF1
SNDQ?   =$DC1F   STXP    =$DC27   INSTART  =$DC2B   PUTCHR  =$DC3F
PENCHR  =$DC4B   PECHR   =$DC4C   SKP     =$DC57   NUMIN   =$DC63
DATIN   =$DC69   WNX     =$DC72   SWPNT   =$DC7E   INPFIN  =$DC99
FINDATA =$DCA0   NXS     =$DCB9   INPDONE  =$DCC6   NTD     =$DCD1
RET11   =$DCDE   EXIG    =$DCDF   REENT   =$DCEF   NEXT    =$DCF9
VARNXT  =$DCFF   SKPV    =$DD02   GERR    =$DD0D   GOTFOR  =$DD0F
GONEWST =$DD52   ENDFOR  =$DD55   FRMNUM  =$DD67   CHKNUM  =$DD6A
CHKSTR  =$DD6C   CHKVAL  =$DD6D   RET12   =$DD73   CV2     =$DD74
MISMTCH =$DD76   JERROR  =$DD78   FRMEVL  =$DD7B   FEVLOOP =$DD86
FRMEVL2 =$DD95   CPROP   =$DD98   CHKTYP  =$DDB4   ARITH   =$DDC5
PREFTEST=$DDCD   NXOP    =$DDD6   SAVOP   =$DDD7   COMPARE =$DDE4
MND     =$DDEE   PREFNC  =$DDF6   PSHMAD  =$DDFD   SNTXERR =$DE0D
PSHF    =$DE10   PSHFACX =$DE15   PUSHFAC =$DE20   NOTMATH =$DE35
GOEX    =$DE38   DOMTH   =$DE3A   DMTH    =$DE41   DOMATH  =$DE43
EXIT    =$DE5D   GETVAL  =$DE60   SKIP    =$DE64   NUMBER  =$DE69
VAR?    =$DE6C   STRTXT  =$DE81   ST1     =$DE8A   NOT?    =$DE90
EQUOP   =$DE98   NOTZ    =$DE9F   FN?     =$DEA4   SGN?    =$DEAB
PARCHK  =$DEB2   CHKCLS  =$DEB8   CHKOPN  =$DEBB   CHKCOM  =$DEBE
SYNCHR  =$DEC0   SYNERR  =$DEC9   MIN     =$DECE   EQUL    =$DED0
VARL    =$DED5   VR1     =$DEE5   VR2     =$DEF6   SCREEN  =$DEF9
UNARY   =$DF0C   NOTinstr=$DF3A   GOROUT  =$DF3F   OR      =$DF4F
AND     =$DF55   FALSE   =$DF5D   TRUE    =$DF60   POSOP   =$DF65
STRCMP  =$DF7D   SFS     =$DFA5   CMPLOOP  =$DFAA   NUMCMP  =$DFB0
DOCMP   =$DFB5   CMPDONE =$DFC1   JF      =$DFCA   PDL     =$DFCD
NXDIM   =$DFD6   DIM     =$DFD9   PTRGET   =$DFE3   PTRGET2 =$DFE8
PTRGET3 =$DFEA   BADNAM  =$DFF4   NAMOK   =$DFF7   MORNAM  =$E007
```

```
GTLT     =$E011   BYPASS   =$E012   STRNG?   =$E01C   INTVAR? =$E026
NIN      =$E036   SCDCH    =$E03D   JARY     =$E046   BSB     =$E049
VSEARCH  =$E04F   NXVAR    =$E059   NV1      =$E05B   NV2     =$E065
NXPTR    =$E073   ISLETC   =$E07D   RTN1     =$E086   NOTFND  =$E087
TWOBRK   =$E09A   NEWVAR   =$E09C   NWV      =$E0B2   SETVPNT =$E0DE
SVP      =$E0E8   GETARY   =$E0ED   GETARY2  =$E0EF   GD      =$E0F9
NEGNUM   =$E0FE   MAKINT   =$E102   MKINT    =$E108   AYINT   =$E10C
MI1      =$E119   MI2      =$E11B   ARRAY    =$E11E   NXTDIM  =$E12C
FNDARY   =$E169   ARYLOOP  =$E16D   ARYNAM?  =$E179   NXARY   =$E188
SUBERR   =$E196   IQERR    =$E199   JER      =$E19B   ARYFOUND=$E19E
CHKDIM   =$E1AA   NOTFOUND=$E1B8   MAKARY   =$E1C1   NINT    =$E1D5
RAR      =$E1DE   SAVDIM   =$E1E7   DFLTDIM  =$E1F7   ZARY    =$E21A
ZLUP     =$E229   NXPG     =$E22E   FNDELEM  =$E24B   DIMLUP  =$E253
GSE      =$E269   GME      =$E26C   DIMOK    =$E26F   DIMOK2  =$E270
NXDM     =$E281   NINTA    =$E292   RARY     =$E298   RTN2    =$E2AC
MULT     =$E2AD   MU1      =$E2B6   MU2      =$E2C0   MU3     =$E2D9
FRE      =$E2DE   FRE2     =$E2E5   GIVAYF   =$E2F2   POS     =$E2FF
SNGFLT   =$E301   ERRDIR   =$E306   UNDFNC   =$E30E   DEF     =$E313
FNC?     =$E341   FUNCT    =$E354   SAVOLD   =$E378   GETOLD  =$E3A9
FNCDATA  =$E3AF   STR      =$E3C5   STRINI   =$E3D5   STRSPA  =$E3DD
STRLIT   =$E3E7   STRLT2   =$E3ED   FEND     =$E3F7   QUO?    =$E404
ZEND     =$E408   NZ       =$E409   FE1      =$E415   FE2     =$E41F
PUTNEW   =$E42A   JERR     =$E432   PUTEMP   =$E435   GETSPA  =$E452
GETSPC   =$E454   CY       =$E45F   GOTSPA   =$E469   FULL    =$E474
GARBAG   =$E484   FNDVAR   =$E488   TVAR     =$E4A0   SVARS   =$E4A9
SVAR     =$E4B5   SVARGO   =$E4BD   ARYVAR   =$E4C2   ARYVA2  =$E4CA
ARYVA3   =$E4CE   ARYVGO   =$E4D9   ARYSTR   =$E50C   GOGO    =$E514
DVARS    =$E519   DVAR     =$E523   DV1      =$E538   DV2     =$E542
DVARTS   =$E552   VDONE    =$E55D   GRBPAS   =$E562   CAT     =$E597
NTL      =$E5B7   MOVINS   =$E5D4   MOVSTR   =$E5E2   MOVESTR =$E5E6
MVS2     =$E5EA   MVS3     =$E5F3   FRESTR   =$E5FD   FREFAC  =$E600
FRETMP   =$E604   NB       =$E630   FRETMS   =$E635   RTN3    =$E645
CHRSTR   =$E646   LEFTSTR  =$E65A   INS1     =$E660   INS2    =$E666
INS3     =$E667   INS4     =$E668   M BPD     =$E67F   RIGHTSTR=$E686
MIDSTR   =$E691   INS5     =$E6A2   INSTRNG  =$E6B9   LEN     =$E6D6
GETSTR   =$E6DC   ASC      =$E6E5   GOIQ     =$E6F2   GTBYTC  =$E6F5
GETBYT   =$E6F8   CONINT   =$E6FB   VAL      =$E707   VL2     =$E70F
VL3      =$E727   POINT    =$E73D   GTNUM    =$E746   COMBYTE =$E74C
GETADR   =$E752   PEEK     =$E764   POKE     =$E77B   WAIT    =$E784
WT2      =$E793   WT3      =$E797   RTN4     =$E79F   FADDH   =$E7A0
FSUB     =$E7A7   FSUBT    =$E7AA   AD0      =$E7B9   FADD    =$E7BE
FADDT    =$E7C1   AD1      =$E7C6   AD2      =$E7CE   AD3     =$E7EA
AD4      =$E7EE   AD5      =$E7FA   SUBMAN   =$E806   SGNIF   =$E829
SIGNIF   =$E82E   FLOOP    =$E832   ZEROFAC  =$E84E   AtoFAC  =$E850
AtoFACS  =$E852   ADMAN    =$E855   FR1      =$E874   FR2     =$E880
FR3      =$E88D   FROUND   =$E88F   RTN5     =$E89D   NEGFAC  =$E89E
NEG2     =$E8A4   PLUSEPS  =$E8C6   RTN6     =$E8D4   OVERFLOW=$E8D5
SHFTRES  =$E8DA   NXSFT    =$E8DC   SHIFT    =$E8F0   SH1     =$E8FD
SH2      =$E903   SHFTR    =$E907   SH3      =$E911   ONE     =$E913
LOGSER   =$E918   SQRhalf  =$E92D   SQRtwo   =$E932   HALFneg =$E937
LOGtwo   =$E93C   LOG      =$E941   GIQ      =$E948   LG2     =$E94B
FMULT    =$E97F   FMULTT   =$E982   FMU      =$E987   FM1     =$E9B0
FM2      =$E9B5   FM3      =$E9B8   FM4      =$E9D4   RTN7    =$E9E2
CONUPK   =$E9E3   ADEXP    =$EA0E   ADEX2    =$EA10   ADEX3   =$EA1B
ADEX4    =$EA26   OUTOFRNG=$EA2B   ZERO     =$EA31   JOV     =$EA36
MUL10    =$EA39   RTN8     =$EA4F   NUM10    =$EA50   DIV10   =$EA55
DIV      =$EA5E   FDIV     =$EA66   FDIVT    =$EA69   FD1     =$EA80
```

```
FD2       =$EA96    FD3       =$EAA3    FD4       =$EAA6    FD5       =$EAB4
FD6       =$EAD1    FD7       =$EAD5    DIVZ      =$EAE1    RES>FAC   =$EAE6
MOVFM     =$EAF9    MOV2F     =$EB1E    MOV1F     =$EB21  ? MOVML     =$EB23
SETFOR    =$EB27    MOVMF     =$EB2B    MOVFA     =$EB53    MFA       =$EB55
MFA2      =$EB59    MOVAF     =$EB63    MAF       =$EB66    MAF2      =$EB68
RTN9      =$EB71    RNDB      =$EB72    ROUND     =$EB7A    SIGN      =$EB82
SIGN1     =$EB86    SIGN2     =$EB88    RTN10     =$EB8F    SGN       =$EB90
FLOAT     =$EB93    FLO1      =$EB9B    FLO2      =$EBA0    ABS       =$EBAF
FCOMP     =$EBB2    FCOMP2    =$EBB4    FC1       =$EBE9    FC2       =$EBEF
QINT      =$EBF2    QI1       =$EC06    RTN11     =$EC11    QI2       =$EC12
INT       =$EC23    ZFAC      =$EC40    RTN12     =$EC49    FIN       =$EC4A
FIN2      =$EC4E    FIN3      =$EC5D    EVAL      =$EC61    NXDIGIT   =$EC64
CHKDP     =$EC66    SETSGN    =$EC85    DPDIG     =$EC87    GOGEX     =$EC8A
SGNCHK    =$EC8C    SETDP     =$EC98    ADJEXP    =$EC9E    AEX       =$ECA0
DPLEFT    =$ECA9    DPRIGHT   =$ECB2    EVDONE    =$ECB9    EVD       =$ECBE
INSRTDIG  =$ECC1    NDP       =$ECC8    ADDACC    =$ECD5    GETEXP    =$ECE8
MVDG      =$ECF7    STEX      =$ED05    HMmiTNTH  =$ED0A    BILmiONE  =$ED0F
BILLION   =$ED14    INPRT     =$ED19    LINPRT    =$ED24  ? PRNTFAC   =$ED2E
PRSTR     =$ED31    FOUT      =$ED34    FACSTRNG  =$ED36    SFSG      =$ED41
NOTZE     =$ED4F    MB        =$ED57    STE       =$ED60    CMPBM1    =$ED62
CMPHM     =$ED6D    JM10      =$ED78    JD10      =$ED7F    ROUN      =$ED86
INTPART   =$ED89    DPLOC     =$ED9E    DPL       =$ED9F    PUTDP     =$EDAA
SVY       =$EDBB    MAKSTR    =$EDBD    MSLUP     =$EDC1    PARITY?   =$EDE5
COUNTED   =$EDE7    MAKDIGIT  =$EDEE    SAVY      =$EE09    MVBACK    =$EE19
NEEDEX?   =$EE26    PUTEX     =$EE36    WHATX     =$EE42    WNDUP     =$EE57
MARKEND   =$EE5A    PNTSTK    =$EE5F    HALF      =$EE64    DECTBL    =$EE69
TEND      =$EE8D    SQR       =$EE8D    FPWRT     =$EE97    PW1       =$EEA0
PW2       =$EEBA    NEGOP     =$EED0    RTN13     =$EEDA    LOGe      =$EEDB
EXPSER    =$EEE0    EXP       =$EF09    X1        =$EF19    QQR       =$EF24
X2        =$EF27    X3        =$EF37    ODDSER    =$EF5C    SERIES    =$EF72
SERMAIN   =$EF76    SS        =$EF85    SERLOOP   =$EF89    NXTERM    =$EF96
RTN14     =$EFA5    RNDADJ1   =$EFA6    RNDADJ2   =$EFAA    RND       =$EFAE
RD1       =$EFCC    RD2       =$EFE7    COS       =$EFEA    SIN       =$EFF1
SI1       =$F023    SI2       =$F026    SI3       =$F033    TAN       =$F03A
TAN2      =$F062    PIhalf    =$F066    PIdoub    =$F06B    QUARTER   =$F070
SINSER    =$F075    ATN       =$F09E    ATN1      =$F0A6    ATN2      =$F0B4
ATN3      =$F0C7    RTN15     =$F0CD    ATNSER    =$F0CE    ZPSTUFF   =$F10B
RTN16     =$F122    COLDST    =$F128    MVZP      =$F152    FNDMEMHI  =$F181
MEMFOUND  =$F195    CALL      =$F1D5    INNU      =$F1DE    PRNU      =$F1E5
PLOTFNS   =$F1EC    GOERR     =$F206    LINCOOR   =$F209    AT?       =$F218
PLOT      =$F225    HLIN      =$F232    VLIN      =$F241    COLOR     =$F24F
VTAB      =$F256    SPEED     =$F262    TRACE     =$F26D    NOTRACE   =$F26F
NORMAL    =$F273    INVERSE   =$F277    NRM       =$F279    SI        =$F27B
FLASH     =$F280    HIMEM     =$F286    JMM       =$F296    SETHI     =$F299
LOMEM     =$F2A6    ONERR     =$F2CB    HANDLERR  =$F2E9    RESUME    =$F318
JSYN      =$F32E    DEL       =$F331    MOVDWN    =$F365    MOVIT     =$F367
NDX       =$F388    GR        =$F390    TEXT      =$F399    STORE     =$F39F
SLI       =$F3AF    RECALL    =$F3BC    MOK       =$F3D2    HGR2      =$F3D8
HGR       =$F3E2    SETHPG    =$F3EA  ? HCLR      =$F3F2  ? BKGND     =$F3F6
BKGND1    =$F3FE    HPOSN     =$F411    HPOSN1    =$F441    HPOSN2    =$F442
HPLOT0    =$F457    LFTRT     =$F465  ? LEFT      =$F467    LR1       =$F46E
LEFT1     =$F471    LEFT2     =$F476    NEWNDX    =$F478  ? CSHIFT    =$F47C
CSHFT2    =$F47E    RTS1      =$F489    RIGHT     =$F48A    LRUDX1    =$F49C
LRUDX2    =$F49D    LRUD1     =$F4B3    LRUD2     =$F4B4    LRUD3     =$F4C4
LRUD4     =$F4C8    EQ3       =$F4CD    UPDOWN    =$F4D3  ? UP        =$F4D5
UP1       =$F4EB    UP5       =$F4F6    UP2       =$F4FB    UP3       =$F4FD
UP4       =$F4FF    UPDWN1    =$F501    DOWN      =$F505    EQ4       =$F508
```

```
  DOWN3    =$F524    DOWN1    =$F52A      DOWN2    =$F52C  ?  HLINRL   =$F530
  HGLIN    =$F53A    HLIN2    =$F550      HLIN3    =$F568     MOVEX    =$F57C
  MOVEX2   =$F581    HCOUNT   =$F58B      HLIN4    =$F59E     MSKTBL   =$F5B2
  EQ1C     =$F5B9    COSTBL   =$F5BA   ?  HFIND    =$F5CB     HFIND1   =$F5F0
  HFIND2   =$F5FE    RTS2     =$F600   ?  DRAWO    =$F601     DRAW1    =$F605
  DRAW2    =$F626    DRAW3    =$F630      DRAW4    =$F63D     DRAW5    =$F648
?  XDRAWO   =$F65D    XDRAW1   =$F661      XDRAW2   =$F682     XDRAW3   =$F68C
  XDRAW4   =$F699    XDRAW5   =$F6A4   M  NI       =$F6B4     HFNS     =$F6B9
  HFNS1    =$F6CD    GGERR    =$F6E6      HCOLOR   =$F6E9     RTS3     =$F6F5
  COLORTBL=$F6F6     HPLOT    =$F6FE      HP2      =$F708     HP3      =$F70F
  ROT      =$F721    SCALE    =$F727      DRWPNT   =$F72D     DP1      =$F741
  DP2      =$F747    DR3      =$F766      DRAW     =$F769     XDRAW    =$F76F
  SHLOAD   =$F775    SL1      =$F796      SL2      =$F7A0     SL3      =$F7A3
  TAPEPNT  =$F7BC    GETARYPT=$F7D9       HTAB     =$F7E7     HTAB1    =$F7EC
  HTAB2    =$F7FA    MONPLOT  =$F800      HLINE    =$F819     VLINE    =$F828
  SETCOL   =$F864    SCRN     =$F871      PREAD    =$FB1E     SETTXT   =$FB39
  SETGR    =$FB40    TABV     =$FB5B      HOME     =$FC58     MONWAIT  =$FCA8
  RD2BIT   =$FCFA    RDKEY    =$FD0C      GETLN    =$FD6A     COUT     =$FDED
  INPORT   =$FE8B    OUTPORT  =$FE95      WRITE    =$FECD     MONREAD  =$FEFD
  MONREAD2=$FF02
```

XREF

Cross referenced symbol table - alphabetical order:

```
    A1H       =$3D     121.76*    121.1168   124.781    124.808
    A1L       =$3C     121.75*    121.1167   121.1175   121.1175   124.784
                       124.814    124.823    124.823
    A2H       =$3F     121.78*    121.1171   124.782    124.800
    A2L       =$3E     121.77*    121.1170   121.1176   121.1176   124.786
                       124.792    124.819    124.819
    ABS       =$EBAF   121.272    123.515*
MD  AD        =$D000   121.880    121.881    121.1156   121.1156   121.1331
                       121.1334   122.1101   122.1102   123.101    123.102
                       123.103    123.104    123.251    123.252    123.253
                       123.254    124.668    124.819    124.819
    AD0       =$E7B9   123.29*    123.56
    AD1       =$E7C6   123.32     123.34*
    AD2       =$E7CE   123.38*    123.327
    AD3       =$E7EA   123.43     123.53*
    AD4       =$E7EE   123.52     123.55*
    AD5       =$E7FA   123.30     123.42     123.61*
MD  ADD       =$D000   121.1156   124.819
    ADDACC    =$ECD5   123.218    123.683    123.688*
    ADDON     =$D998   121.1247*  121.1285   121.1635   124.158
    ADEX2     =$EA10   123.293*   123.1000
    ADEX3     =$EA1B   123.296    123.300*
    ADEX4     =$EA26   123.303    123.305*
    ADEXP     =$EA0E   123.225    123.292*   123.351
    ADJEXP    =$EC9E   123.626    123.643    123.657*
    ADMAN     =$E855   123.62     123.99*
    AEX       =$ECA0   123.647    123.658*
    AMPER     =$03F5   121.176*   121.253
    AND       =$DF55   121.309    122.275*
    ARG       =$A5     121.128*   121.130    122.132    122.132    122.133
                       122.133    122.134    122.134    122.272    122.275
                       122.288    122.289    122.290    122.302    122.303
                       122.305    122.306    122.325    123.36     123.37
                       123.64     123.66     123.101    123.102    123.103
                       123.104    123.251    123.252    123.253    123.254
                       123.271    123.274    123.277    123.285    123.288
                       123.292    123.356    123.359    123.362    123.365
                       123.377    123.378    123.379    123.380    123.385
                       123.385    123.386    123.386    123.387    123.387
                       123.388    123.388    123.463    123.473    123.919
                       123.984    123.987
    ARGSGN    =$AA     121.130*   122.286    123.24     123.45     123.280
                       123.283    123.460    123.692    123.926    123.1080
    ARITH     =$DDC5   122.55     122.57*
    ARRAY     =$E11E   122.400    122.518*
    ARYFOUND=$E19E     122.566    122.581*
    ARYLOOP   =$E16D   122.553*   122.575
    ARYNAM?   =$E179   122.556    122.559*
    ARYPNT    =$94     121.110*   122.461    122.462    122.464    122.465
                       122.497    122.498    122.644    122.646    122.649
                       122.661    122.663    122.666    122.729    122.732
                       122.1011   122.1012   122.1015   122.1016   122.1033
                       122.1034   122.1037   122.1038   122.1050   122.1052
                       122.1103   122.1103   122.1107   122.1110   122.1111
    ARYSTR    =$E50C   122.1050*  122.1055
```

MD = macro def.

	ARYTAB	=$6B	121.88*	121.860	121.860	122.411	122.413
			122.455	122.455	122.467	122.468	122.551
			122.552	122.1004	122.1006		
	ARYVA2	=$E4CA	122.1015*	122.1040	122.1042		
	ARYVA3	=$E4CE	122.1017*	122.1053			
	ARYVAR	=$E4C2	122.1007	122.1011*			
	ARYVGO	=$E4D9	122.1018	122.1020	122.1023*		
	ASC	=$E6E5	121.290	122.1299*			
	ASCHEX	=$DA12	121.1312*	121.1338			
	AT?	=$F218	124.54	124.59*			
	ATN	=$F09E	121.285	123.1142*			
	ATN1	=$F0A6	123.1144	123.1146*			
	ATN2	=$F0B4	123.1149	123.1153*			
	ATN3	=$F0C7	123.1158	123.1162*			
	ATNSER	=$F0CE	123.1153	123.1154	123.1167*		
	AYINT	=$E10C	121.1354	122.507*			
	AtoFAC	=$E850	123.95*	123.921			
	AtoFACS	=$E852	123.96*	123.304			
	BADNAM	=$DFF4	122.357*	122.386			
	BADSUBS	=$D2CB	121.437*	122.576			
	BILLION	=$ED14	123.720*	123.760	123.761		
	BILmiONE	=$ED0F	123.719*	123.765	123.766		
?	BKGND	=$F3F6	124.267*				
	BKGND1	=$F3FE	124.271*	124.275	124.279		
	BLTU	=$D393	121.481*	121.648	122.463		
	BLTU2	=$D39A	121.488*	122.1104			
M	BPD	=$E67F					
	BREAKIN	=$D35D	121.452*	121.1132	121.1133		
	BSB	=$E049	122.399	122.401*			
	BUMP	=$D000	121.1063	121.1249	122.1048	122.1086	122.1162
			122.1190	122.1236			
	BYPASS	=$E012	122.374*	122.375	122.377		
	CALL	=$F1D5	121.218	124.18*			
	CANTCON	=$D332	121.444*	121.1139			
	CAT	=$E597	122.56	122.1116*			
	CH	=$24	121.67*	121.930	121.935	121.1429	121.1436
			121.1447	122.791	124.848		
	CHARAC	=$0D	121.51*	121.1255	121.1259	121.1260	121.1314
			121.1334	121.1577	121.1579	121.1583	122.897
			122.907	123.597	123.934	123.976	
	CHKCLS	=$DEB8	122.188*	122.233	122.547	122.811	122.1272
	CHKCOM	=$DEBE	121.1612	122.192*	122.247	122.343	122.1252
			122.1344				
	CHKDIM	=$E1AA	122.585	122.589*			
	CHKDP	=$EC66	123.620	123.623*			
	CHKMEM	=$D3D6	121.520*	121.1003	121.1194	122.30	
	CHKNUM	=$DD6A	121.1015	121.1712*	122.65	122.123	122.270
			122.810	122.832	122.837	122.878	
	CHKOPN	=$DEBB	122.186	122.190*	122.245	122.806	
	CHKSTR	=$DD6C	121.1714*	122.248	122.1118	122.1165	
	CHKTYP	=$DDB4	122.37	122.39	122.49*		
	CHKVAL	=$DD6D	121.1349	121.1715*	122.284		
	CHRGET	=$B1	121.137*	121.585	121.898	121.1032	121.1075
			121.1091	121.1302	121.1337	121.1453	121.1569
			121.1708	122.47	122.145	122.196	122.226
			122.242	122.369	122.374	122.394	122.503
			122.1307	123.621	123.627	123.640	123.1221

? = not referenced
m = used in macro

CHRGOT =$B7 121.138* 121.892 121.1029 121.1199 121.1277
121.1287 121.1404 121.1553 121.1603 121.1610
121.1705 122.34 122.346 122.351 122.354
122.543 122.863 122.1249 122.1313 122.1334
122.1380 124.172 124.726 124.766
CHRSTR =$E646 121.291 122.1206*
CLEAR =$D66A 121.267 121.858*
CLEARC =$D66C 121.859* 121.1184 124.149
CMDTABL =$D000 121.206* 121.1087 121.1089
CMPBM1 =$ED62 123.765* 123.780
CMPDONE =$DFC1 122.322 122.324 122.329 122.331*
CMPHM =$ED6D 123.770* 123.777
CMPLOOP =$DFAA 122.318* 122.327
MD CMPR =$D000 124.130 124.144 124.146 124.196 124.204
COLCOUNT=$EA 121.161* 124.377 124.389 124.602 124.654
COLDST =$F128 122.365 123.1198* 123.1203 123.1204
COLON? =$D842 121.1051 121.1093*
COLOR =$F24F 121.238 124.89*
COLORTBL=$F6F6 124.716 124.720*
COMBYTE =$E74C 122.1344* 122.1382
COMPARE =$DDE4 122.50 122.75*
CON =$D8A1 121.1141 121.1143*
CONINT =$E6FB 122.339 122.1206 122.1309*
CONT =$D896 121.265 121.1138*
CONUPK =$E9E3 123.22 123.31 123.221 123.267* 123.344
COPSTR =$DA9A 121.1367 121.1372 121.1375 121.1380*
COPY =$DAB7 121.1382 121.1390*
COS =$EFEA 121.282 123.1073*
COSTBL =$F5BA 124.538* 124.593 124.597 124.645 124.649
COUNTED =$EDE7 123.828 123.830*
COUNTH =$1D 121.66* 124.493 124.512
COUT =$FDED 121.190* 121.1483
CPRMASK =$16 121.61* 122.131 122.334
CPROP =$DD98 122.35* 122.48
CPRTYP =$89 121.104* 122.33 122.43 122.44 122.46
122.49 122.80 122.92 122.297
CRDO =$DAFB 121.562 121.579 121.913 121.933 121.1405
121.1424* 121.1432 123.1232 124.845
CSHFT2 =$F47E 124.273 124.324 124.351*
? CSHIFT =$F47C 124.350*
CTRC? =$D86C 121.1114 121.1116*
CURLIN =$75 121.93* 121.574 121.589 121.1012 121.1012
121.1044 121.1058 121.1061 121.1066 121.1071
121.1072 121.1124 121.1129 121.1129 121.1146
121.1146 121.1180 121.1196 121.1196 121.1205
121.1242 121.1244 121.1498 121.1499 121.1693
121.1695 122.796 123.725 123.726 123.1199
124.156 124.156 124.168 124.168 124.171
124.171 124.176 124.176
CURLSV =$F6 121.167* 124.156 124.156 124.171 124.171
CV2 =$DD74 121.1716 121.1719*
CY =$E45F 122.961 122.963*
DATA =$D995 121.209 121.1246* 122.817
DATAFLG =$13 121.57* 121.58 121.709 121.719 121.777
DATAN =$D9A3 121.1004 121.1246 121.1252* 121.1618
DATIN =$DC69 121.1599* 121.1615
DATLIN =$7B 121.96* 121.1496 121.1497 121.1628 121.1632

	DATPTR	=$7D	121.97*	121.1103	121.1104	121.1540	121.1541
MD	DECR	=$D000	122.23	122.78			
	DECTBL	=$EE69	123.814	123.817	123.820	123.823	123.855
			123.901*				
	DEF	=$E313	121.262	122.804*			
	DEL	=$F331	121.211	124.184*			
	DESC?	=$DA8C	121.1368	121.1373*			
	DEST	=$60	121.84*	121.603	121.610	121.622	121.629
			121.633	121.1676	122.1323	122.1328	122.1330
			122.1333	122.1338	123.520	123.521	123.523
			123.527	123.532	123.537	123.541	123.547
			124.188	124.188	124.196	124.196	124.201
			124.203	124.203	124.206	124.207	
	DFLTDIM	=$E1F7	122.626	122.633*			
	DIM	=$DFD9	121.212	122.344*			
	DIMFLG	=$10	121.55*	122.352	122.520	122.550	122.582
			122.625	122.676			
	DIMLUP	=$E253	122.683*	122.715			
	DIMOK	=$E26F	122.691	122.701*			
	DIMOK2	=$E270	122.696	122.702*			
	DIR?	=$DBC7	121.1527	121.1529*			
	DIRCT	=$D7E5	121.1046	121.1049*			
	DIV	=$EA5E	123.340*	123.1081			
	DIV10	=$EA55	123.336*	123.663	123.778		
	DIVZ	=$EAE1	123.345	123.399*			
	DIVbyZRO	=$D2E5	121.439*	123.399			
	DMTH	=$DE41	122.122	122.124*			
	DOCMP	=$DFB5	122.320	122.325*			
	DOERRMSG	=$D419	121.559	121.562*			
	DOMATH	=$DE43	122.73	122.84	122.129*		
	DOMTH	=$DE3A	122.64	122.121*			
	DONE	=$D610	121.771	121.802*			
	DOREENT	=$DB90	121.1503	121.1506*			
	DOSPC	=$DB35	121.1452	121.1455*			
	DOWN	=$F505	124.399	124.426*			
	DOWN1	=$F52A	124.432	124.445*			
	DOWN2	=$F52C	124.436	124.444	124.446*		
	DOWN3	=$F524	124.440	124.442*			
	DP1	=$F741	124.751	124.753*			
	DP2	=$F747	124.754	124.757*			
	DP3	=$F766	124.768	124.772*			
	DPDIG	=$EC87	123.634	123.636	123.640*	123.716	
	DPFLG	=$9B	121.123*	123.648	123.649	123.676	
	DPL	=$ED9F	123.789	123.794*			
	DPLEFT	=$ECA9	123.663*	123.665			
	DPLOC	=$ED9E	123.787	123.793*			
	DPRIGHT	=$ECB2	123.662	123.667*	123.669		
	DRAW	=$F769	121.226	124.775*			
?	DRAW0	=$F601	124.583*				
	DRAW1	=$F605	124.585*	124.776			
	DRAW2	=$F626	124.604*	124.627	124.630		
	DRAW3	=$F630	124.609*	124.622			
	DRAW4	=$F63D	124.613	124.616*			
	DRAW5	=$F648	124.619	124.621*			
	DRWPNT	=$F72D	124.746*	124.775	124.777		
	DSCLEN	=$8F	121.107*	122.1002	122.1014	122.1083	122.1084
			123.1231				

DSCPTR	=$8C	121.106*	121.1386	121.1386	121.1390	121.1391
		121.1395	121.1399	122.889	122.889	122.1130
		122.1131	122.1219	122.1222	122.1229	122.1230
		122.1243	122.1260	122.1281	122.1281	
DSCTMP	=$9D	121.115*	124.703	124.708	124.731	124.734
DV1	=$E538	122.1071	122.1075*			
DV2	=$E542	122.1077	122.1080*			
DVAR	=$E523	122.998	122.1054	122.1063*		
DVARS	=$E519	122.1008	122.1057*			
DVARTS	=$E552	122.1058	122.1061	122.1064	122.1072	122.1074
		122.1076	122.1079	122.1084*		
DXH	=$D1	121.142*	124.370	124.382	124.392	124.471
		124.492	124.522	124.604	124.623	124.656
		124.673				
DXL	=$D0	121.141*	124.474	124.489	124.519	124.594
		124.611	124.646	124.663		
DY	=$D2	121.143*	124.485	124.500	124.599	124.617
		124.651	124.668			
EH	=$D5	121.146*	124.472	124.502	124.504	124.521
		124.607	124.616	124.618	124.659	124.668
		124.668				
EL	=$D4	121.145*	124.475	124.499	124.501	124.518
		124.520	124.606	124.609	124.612	124.658
		124.661	124.664			
END	=$D870	121.206	121.1120*			
END2	=$D871	121.1118	121.1121*			
END3	=$D888	121.1126	121.1130*			
END4	=$D88A	121.1079	121.1132*			
ENDCHR	=$0E	121.52*	121.723	121.781	121.784	121.1257
		121.1258	121.1261	121.1264	121.1586	122.898
		122.909				
ENDFOR	=$DD55	121.1691	121.1701*			
ENDRNG	=$D6C4	121.895	121.898*			
EQ1C	=$F5B9	124.402	124.429	124.533*		
EQ3	=$F4CD	124.395*	124.406			
EQ4	=$F508	124.428*	124.435			
EQUAL	=$D0CA	121.314*	122.168			
EQUL	=$DED0	122.169	122.202*			
EQUOP	=$DE98	121.315	122.171*			
ERFLG?	=$D863	121.1112*	121.1536			
ERLIN	=$DB7F	121.1495	121.1498*			
ERLUP	=$D41F	121.564*	121.569			
ERRDIR	=$E306	121.1512	121.1529	122.796*	122.805	
ERRFLG	=$D8	121.148*	121.558	121.584	121.1113	121.1502
		124.155				
ERRIN	=$D350	121.447*	121.571	121.572		
ERRLIN	=$DA	121.149*	124.168	124.168	124.176	124.176
ERRMSG	=$D260	121.428*	121.557	121.564	121.1139	121.1231
		121.1233	121.1622	121.1666	121.1720	122.198
		122.576	122.578	122.581	122.598	122.799
		122.801	122.934	122.976	122.1125	123.147
		123.399				
ERRNUM	=$DE	121.151*	124.162			
ERROR	=$D412	121.275	121.558*	121.1142	121.1234	121.1721
		122.199	122.579	122.599	122.802	122.935
		122.1126	123.148	123.400		
ERRPOS	=$DC	121.150*	124.169	124.169	124.177	124.177

ERRSTK =$DF 121.152* 124.164 124.178
EVAL =$EC61 123.618 123.621* 123.650 123.684
EVD =$ECBE 123.671 123.673*
EVDONE =$ECB9 123.661 123.666 123.670*
EXIG =$DCDF 121.1647 121.1648 121.1652*
EXIT =$DE5D 122.120 122.137*
EXP =$EF09 121.281 123.918 123.942 123.962*
EXPON =$9A 121.122* 123.646 123.657 123.660 123.664
123.668 123.698 123.708 123.715 123.795
123.866 123.871
EXPSER =$EEE0 123.952* 123.994 123.995
EXPSGN =$9C 121.124* 123.639 123.642 123.702
EXTRAFAC=$AC 121.132* 123.34 123.54 123.58 123.70
123.88 123.90 123.100 123.108 123.127
123.135 123.135 123.136 123.152 123.166
123.231 123.259 123.396 123.427 123.457
123.467 123.476 123.484 123.511 123.546
123.589 123.965 123.991 123.1065
EXTRASV =$92 121.117* 123.35 123.50 123.69 123.99
123.969 123.990
FAC =$9D 121.125* 121.129 121.535 121.542 121.544
121.737 121.739 121.747 121.753 121.792
121.795 121.798 121.953 121.954 121.963
121.965 121.1019 121.1020 121.1282 121.1298
121.1388 122.112 122.112 122.113 122.113
122.114 122.137 122.171 122.273 122.277
122.299 122.300 122.301 122.309 122.313
122.326 122.507 122.538 122.540 122.687
122.689 122.708 122.712 122.786 122.787
122.889 122.889 122.891 122.892 122.893
122.901 122.902 122.914 122.937 122.939
122.941 122.944 122.945 122.1116 122.1116
122.1123 122.1166 122.1167 122.1213 122.1248
122.1263 122.1265 122.1310 122.1312 122.1347
122.1351 122.1352 123.26 123.41 123.44
123.51 123.63 123.71 123.72 123.73
123.74 123.80 123.82 123.83 123.84
123.85 123.86 123.87 123.89 123.95
123.101 123.101 123.102 123.102 123.103
123.103 123.104 123.104 123.109 123.110
123.111 123.112 123.115 123.119 123.121
123.123 123.124 123.125 123.126 123.131
123.131 123.132 123.132 123.133 123.133
123.134 123.134 123.138 123.140 123.142
123.144 123.197 123.201 123.233 123.235
123.237 123.239 123.289 123.295 123.302
123.328 123.349 123.350 123.352 123.357
123.360 123.363 123.366 123.385 123.386
123.387 123.388 123.401 123.401 123.402
123.402 123.412 123.415 123.418 123.423
123.426 123.441 123.444 123.447 123.452
123.455 123.464 123.472 123.482 123.490
123.500 123.502 123.504 123.508 123.509
123.510 123.530 123.534 123.538 123.542
123.548 123.557 123.568 123.578 123.579
123.580 123.585 123.595 123.596 123.600
123.601 123.602 123.603 123.695 123.727

		123.728	123.753	123.812	123.815	123.816
		123.818	123.819	123.821	123.822	123.824
		123.946	123.971	123.985	123.986	123.1058
		123.1059	123.1060	123.1061	123.1064	123.1067
		123.1146				
FACSGN	=$A2	121.129*	121.1017	121.1681	122.97	122.135
		122.315	122.321	122.505	123.23	123.23
		123.46	123.96	123.130	123.130	123.281
		123.306	123.309	123.421	123.450	123.461
		123.492	123.512	123.515	123.528	123.550
		123.561	123.576	123.590	123.591	123.693
		123.745	123.749	123.948	123.948	123.1063
		123.1090	123.1094	123.1119	123.1142	
FACSTRNG	=$ED36	122.880	123.743*			
FADD	=$E7BE	121.1684	123.20	123.31*	123.204	123.216
		123.1034	123.1057	123.1075	123.1100	
FADDH	=$E7A0	123.18*	123.781	123.1093		
FADDT	=$E7C1	121.299	123.27	123.32*	123.696	
FALSE	=$DF5D	122.276	122.279*			
FC1	=$EBE9	123.531	123.535	123.539	123.543	123.550*
FC2	=$EBEF	123.551	123.553*			
FCOMP	=$EBB2	122.292	122.512	123.520*	123.767	123.772
		123.931				
FCOMP2	=$EBB4	121.1687	123.521*			
FD1	=$EA80	123.356*	123.382			
FD2	=$EA96	123.358	123.361	123.364	123.367*	123.381
		123.383				
FD3	=$EAA3	123.369	123.375*	123.392		
FD4	=$EAA6	123.377*	123.390			
FD5	=$EAB4	123.376	123.384*			
FD6	=$EAD1	123.372	123.391*			
FD7	=$EAD5	123.373	123.393*			
FDIV	=$EA66	123.207	123.344*	123.1124	123.1152	
FDIVT	=$EA69	121.305	123.342	123.345*		
FE1	=$E415	122.919	122.921*			
FE2	=$E41F	122.923	122.926*			
FEND	=$E3F7	122.904*	122.910			
FEVLOOP	=$DD86	122.26*	122.93			
FIN	=$EC4A	121.1600	122.147	122.1335	123.609*	
FIN2	=$EC4E	123.611*	123.613			
FIN3	=$EC5D	123.616	123.619*			
FINDATA	=$DCA0	121.1564	121.1618*	121.1637		
FINDEOL	=$D511	121.665*	121.667			
FIRST	=$F0	121.162*	122.229	124.40	124.53	124.55
		124.58	124.68	124.78	124.86	
FL1	=$D61E	121.814*	121.837	121.1217		
FL2	=$D635	121.824	121.827*			
FLASH	=$F280	121.237	124.124*			
FLO1	=$EB9B	122.789	123.504*			
FLO2	=$EBA0	123.507*	123.731			
FLOAT	=$EB93	122.337	123.500*	123.691		
FLOOP	=$E832	123.80*	123.93			
FM1	=$E9B0	123.232	123.234	123.236	123.238	123.243*
FM2	=$E9B5	123.240	123.243	123.246*		
FM3	=$E9B8	123.248*	123.262			
FM4	=$E9D4	123.249	123.255*			
FMU	=$E987	123.222	123.225*			

FMULT =$E97F 123.221* 123.762 123.941 123.964 123.1007
123.1011 123.1025 123.1054
FMULTT =$E982 121.303 123.222*
FN? =$DEA4 122.167 122.178*
FNC? =$E341 122.804 122.825* 122.834
FNCDATA =$E3AF 122.818 122.868*
FNCNAM =$8A 121.105* 122.830 122.831 122.835 122.835
122.838 122.838 122.840 122.844 122.855
122.858 122.862 122.862 122.871 122.875
122.989 122.1082 122.1082 122.1094 122.1101
122.1108 122.1113
FNDARY =$E169 122.519 122.551*
FNDELEM =$E24B 122.594 122.679*
FNDFOR =$D36A 121.460* 121.478
FNDLIN =$D61A 121.597 121.812* 121.891 124.187 124.195
FNDMEMHI=$F181 123.1243* 123.1252
FNDVAR =$E488 122.986* 122.1114
FOR =$D766 121.207 121.991*
FOR2 =$D777 121.995 121.1000*
FORMtoCX=$D31F 121.443* 122.934
FORPNT =$85 121.101* 121.463 121.466 121.468 121.471
121.924 121.927 121.961 121.1036 121.1036
121.1226 121.1341 121.1342 121.1357 121.1360
121.1396 121.1400 121.1549 121.1550 121.1662
121.1663 121.1682 121.1683 122.1378 122.1383
122.1386 122.1387 123.435 123.436
FOUT =$ED34 121.1420 123.732 123.738*
FPGEN =$A4 121.127* 123.159 123.565 123.572 123.582
123.1227
FPWRT =$EE97 121.307 123.918*
FR1 =$E874 123.107* 123.113
FR2 =$E880 123.81 123.113*
FR3 =$E88D 123.105 123.120*
FRE =$E2DE 121.274 122.774*
FRE2 =$E2E5 122.775 122.777*
FREFAC =$E600 121.1458 122.298 122.776 122.1166*
FRESPC =$71 121.91* 122.970 122.971 122.1157 122.1162
122.1162 122.1162
FRESTR =$E5FD 122.1165* 122.1293
FRETMP =$E604 122.304 122.1132 122.1136 122.1168* 122.1231
FRETMS =$E635 121.1392 122.1170 122.1196*
FRETOP =$6F 121.90* 121.529 121.532 121.550 121.553
121.638 121.638 121.859 121.859 121.1366
121.1371 122.779 122.782 122.959 122.960
122.968 122.969 122.986 122.987 122.1070
122.1073 122.1103 122.1103 122.1184 122.1186
122.1190 122.1190 122.1190 123.1258 123.1259
124.136 124.139 124.807 124.813
FRMEVL =$DD7B 121.1276 121.1346 121.1417 121.1711 122.23*
122.187 122.246
FRMEVL2 =$DD95 122.34* 122.1138
FRMNUM =$DD67 121.1016 121.1033 121.1711* 122.504 122.861
122.1308 122.1342 124.18 124.128 124.142
124.685
FROUND =$E88F 123.121* 123.488
FSUB =$E7A7 123.22* 123.210 123.1089 123.1161
FSUBT =$E7AA 121.301 123.23* 123.992 123.1086

FULL =$E474 122.964 122.967 122.976*
FUNCT =$E354 122.180 122.834*
GARBAG =$E484 121.541 122.777 122.979 122.984*
GARFLG =$13 121.58* 122.955 122.977 122.981
GBASH =$27 121.69* 124.295 124.299 124.301 124.304
124.307 124.401 124.423 124.426 124.547
124.555
GBASL =$26 121.68* 124.289 124.292 124.293 124.303
124.329 124.331 124.332 124.375 124.385
124.390 124.391 124.404 124.413 124.417
124.421 124.431 124.437 124.442 124.446
124.505 124.508 124.509 124.545 124.550
GC =$D731 121.952 121.954*
GD =$E0F9 122.495 122.497*
GDBUFS =$D539 121.686 121.688*
GERR =$DD0D 121.1625 121.1667*
GET =$DBA0 121.268 121.1512*
GETADR =$E752 122.1343 122.1347* 122.1361 124.19 124.129
124.143 124.686
GETARY =$E0ED 122.490* 122.589 122.601
GETARY2 =$E0EF 122.491* 124.822
GETARYPT=$F7D9 124.225 124.240 124.828*
GETBYT =$E6F8 121.1292 122.254 122.1253 122.1308* 122.1345
124.25 124.29 124.37 124.43 124.61
124.89 124.95 124.102 124.700 124.713
124.738 124.742 124.746 124.838
GETCHR =$D72C 121.951* 121.969 121.974
GETEXP =$ECE8 123.641 123.698*
GETIN =$D559 121.591 121.595 121.706*
GETLINK =$D63E 121.826 121.832*
GETLN =$FD6A 121.187* 121.684
GETOLD =$E3A9 122.864 122.867*
GETSPA =$E452 122.890 122.955*
GETSPC =$E454 122.956* 122.983
GETSTR =$E6DC 122.1291 122.1293* 122.1299 122.1315
GETVAL =$DE60 122.31 122.143* 122.1117
GGERR =$F6E6 124.691 124.693 124.702 124.711* 124.715
124.752
GIQ =$E948 123.194 123.196*
GIVAYF =$E2F2 122.223 122.784* 122.794
GK =$D860 121.1109 121.1111*
GME =$E26C 122.645 122.652 122.699* 122.758 122.769
GO1 =$D955 121.1207 121.1215*
GO2 =$D959 121.1212 121.1214 121.1217*
GOCMD =$D828 121.1076 121.1080* 121.1290
GOCMD2 =$D82A 121.1081* 121.1301
GOEND =$D826 121.1055 121.1079*
GOERR =$F206 124.39 124.45 124.50* 124.63 124.70
124.77 124.85 124.99 124.711
GOEX =$DE38 122.72 122.120*
GOGEX =$EC8A 123.628 123.641*
GOGO =$E514 122.1051 122.1054*
GOIQ =$E6F2 122.1288 122.1300 122.1305* 122.1311 122.1349
GOLINE =$D935 121.1185 121.1199*
GONEWST =$DD52 121.1700* 121.1707
GOROUT =$DF3F 122.259 122.264*
GOSTART =$D893 121.1134 121.1136*

GOSTROUT=$03 121.49* 123.1207 123.1208 123.1212 123.1279
123.1280
GOSUB =$D921 121.254 121.1193*
GOTFOR =$DD0F 121.1665 121.1668*
GOTO =$D93E 121.249 121.1200 121.1203* 121.1289 124.173
GOTO? =$D9F4 121.1296* 121.1318
GOTSPA =$E469 122.965 122.968*
GOWARM =$00 121.48* 123.1205 123.1206 123.1211 123.1283
123.1284 123.1285
GR =$F390 121.214 124.215*
GRBPAS =$E562 122.1021 122.1094*
GSE =$E269 122.692 122.697*
GSYNER =$D981 121.1236* 121.1297
GTBYTC =$E6F5 121.1439 122.1307*
GTFORPNT=$D365 121.456* 121.994 121.1227 121.1664
GTLT =$E011 122.370 122.373*
GTNUM =$E746 122.1342* 122.1371 122.1377
H2 =$2C 121.70* 124.46 124.56 124.75
HALF =$EE64 123.18 123.19 123.897* 123.914 123.915
HALFneg =$E937 123.190* 123.214 123.215
HANDLERR=$F2E9 121.560 121.1115 121.1505 124.162*
? HCLR =$F3F2 124.265*
HCOLOR =$F6E9 121.224 124.713*
HCOLOR1 =$1C 121.65* 124.266 124.271 124.323 124.328
124.350 124.354 124.356 124.386 124.506
HCOLORZ =$E4 121.156* 124.322 124.717
HCOUNT =$F58B 124.504* 124.523
? HFIND =$F5CB 124.545*
HFIND1 =$F5F0 124.569* 124.571
HFIND2 =$F5FE 124.576 124.578*
HFNS =$F6B9 124.685* 124.724 124.730 124.770
HFNS1 =$F6CD 124.690 124.694*
HGLIN =$F53A 124.457* 124.735
HGR =$F3E2 121.223 124.259*
HGR2 =$F3D8 121.222 124.255*
HIGHDS =$94 121.111* 121.504 121.504 121.506 121.509
121.513 121.515 121.642 121.647 124.823
124.823
HIGHTR =$96 121.112* 121.489 121.491 121.497 121.500
121.502 121.508 121.512 121.514 121.640
121.644 122.456 122.456 122.1101 122.1102
HIMEM =$F286 121.241 124.128*
HIRES =$C057 121.184* 124.263
HISCR =$C055 121.182* 124.255
HLIN =$F232 121.220 124.73*
HLIN2 =$F550 124.464 124.471*
HLIN3 =$F568 124.482 124.485*
HLIN4 =$F59E 124.511 124.514*
HLINE =$F819 121.197* 124.79
? HLINRL =$F530 124.451*
HMASK =$30 121.72* 124.319 124.330 124.338 124.342
124.348 124.359 124.374 124.379 124.387
124.507 124.567
HMmiTNTH=$ED0A 123.718* 123.770 123.771
HNDX =$E5 121.157* 124.316 124.349 124.494 124.561
124.563 124.575 124.600 124.652
HOME =$FC58 121.185* 121.229

HP2	=$F708	124.726*	124.736			
HP3	=$F70F	124.723	124.729*			
HPAG	=$E6	121.158*	124.262	124.267	124.306	124.443
HPLOT	=$F6FE	121.225	124.722*			
HPLOTO	=$F457	124.327*	124.725			
HPOSN	=$F411	124.284*	124.327	124.771		
HPOSN1	=$F441	124.313*	124.315			
HPOSN2	=$F442	124.310	124.314*			
HTAB	=$F7E7	121.228	124.838*			
HTAB1	=$F7EC	124.841*	124.847			
HTAB2	=$F7FA	124.842	124.848*			
IF	=$D9C9	121.251	121.1276*			
IFTRUE	=$D9E1	121.1283	121.1287*			
ILLDIR	=$D2F5	121.440*	122.799			
ILLQUAN	=$D295	121.433*	122.578			
IN	=$0200	121.175*	121.636	121.649	121.649	121.652
		121.689	121.692	121.694	121.698	121.699
		121.718	121.749	121.760	121.769	121.770
		121.782	121.787	121.800	121.802	121.1513
		121.1514	121.1516	121.1531	121.1533	121.1559
		121.1560	121.1561	121.1597	123.1234	123.1235
INCHR	=$D553	121.702*	121.1111			
MD INCR	=$D000	123.1184	123.1268	124.194	124.202	124.203
		124.628	124.678			
INDEX	=$5E	121.83*	121.489	121.499	121.504	121.523
		121.525	121.601	121.602	121.624	121.626
		121.628	121.632	121.657	121.657	121.660
		121.666	121.670	121.673	121.674	121.677
		121.678	121.679	121.1023	121.1024	121.1316
		121.1327	121.1331	121.1464	122.58	122.60
		122.105	122.106	122.108	122.116	122.641
		122.711	122.742	122.994	122.995	122.1003
		122.1003	122.1023	122.1024	122.1026	122.1029
		122.1032	122.1036	122.1044	122.1048	122.1048
		122.1048	122.1049	122.1057	122.1060	122.1063
		122.1066	122.1069	122.1082	122.1082	122.1086
		122.1086	122.1086	122.1087	122.1150	122.1151
		122.1156	122.1168	122.1169	122.1173	122.1176
		122.1179	122.1192	122.1193	122.1236	122.1236
		122.1236	122.1302	122.1319	122.1322	122.1324
		123.267	123.268	123.270	123.273	123.276
		123.279	123.287	123.408	123.409	123.411
		123.414	123.417	123.420	123.425	123.438
		123.439	123.442	123.445	123.448	123.453
		123.456				
INDX	=$99	121.113*	122.749	122.770		
INLIN	=$D52C	121.682*	121.1532	121.1539		
INLIN2	=$D52E	121.581	121.683*			
INMSG	=$D358	121.449*	123.722	123.723		
INNU	=$F1DE	121.217	124.25*			
INPDONE	=$DCC6	121.1611	121.1639*			
INPERR	=$DB86	121.1501*	121.1616			
INPFIN	=$DC99	121.1598	121.1614*			
INPORT	=$FE8B	121.194*	124.27			
INPRT	=$ED19	121.577	123.722*			
INPTR	=$7F	121.98*	121.1546	121.1547	121.1552	121.1552
		121.1608	121.1608	121.1639	121.1640	121.1645

INPUT =$DBB2 121.210 121.1521*
INPUTERR=$DB71 121.1491* 121.1607
INPUTFLG=$15 121.60* 121.1491 121.1545 121.1555 121.1572
121.1614 121.1641
INS1 =$E660 122.1221* 122.1245
INS2 =$E666 122.1221 122.1225*
INS3 =$E667 122.1226* 122.1261
INS4 =$E668 122.1227* 122.1264 122.1266
INS5 =$E6A2 122.1251 122.1254*
INSRTDIG=$ECC1 123.622 123.675*
INSRTLIN=$D4EA 121.652* 121.655
INSTART =$DC2B 121.1554 121.1569* 121.1638
INSTRNG =$E6B9 122.1218 122.1241 122.1254 122.1272*
INT =$EC23 121.271 123.585* 123.928 123.975 123.1083
INTPART =$ED89 123.768 123.782*
INTVAR? =$E026 122.379 122.383*
INVERSE =$F277 121.236 124.118*
INVFLG =$32 121.73* 124.120
IQERR =$E199 122.513 122.578* 122.1305 123.196 123.1215
123.1216 124.50
ISCNTC =$D858 121.912 121.1041 121.1107*
ISLETC =$E07D 122.148 122.355 122.371 122.376 122.432*
ISTOK? =$D590 121.732 121.735*
JARY =$E046 122.400* 122.403
JD10 =$ED7F 123.769 123.778*
JER =$E19B 122.579* 122.583
JERR =$E432 122.935* 122.978
JERROR =$DD78 121.1667 121.1721*
JF =$DFCA 122.335 122.337*
JGOCMD =$D9E9 121.1288 121.1290*
JLNK =$D8ED 121.1161 121.1163*
JM10 =$ED78 123.773 123.775*
JMM =$F296 124.132* 124.145 124.147
JMPADRS =$90 121.108* 122.265 122.267 122.268 123.1213
JOV =$EA36 123.297 123.311 123.315* 123.324 123.329
123.353
JSY =$D846 121.1084 121.1095*
JSYN =$F32E 124.182* 124.184
KEY =$C000 121.177* 121.1107
LASTOP =$87 121.103* 122.69 122.124
LASTPT =$53 121.81* 122.948 122.1196 122.1198 122.1202
123.1228
? LEFT =$F467 124.338*
LEFT1 =$F471 124.340 124.344*
LEFT2 =$F476 124.345 124.347*
LEFTSTR =$E65A 121.292 122.1218*
LEN =$E6D6 121.287 122.1291*
LENGTH =$91 121.109* 122.1083 122.1096 122.1100 122.1105
122.1276 122.1282
LET =$DA46 121.248 121.993 121.1092 121.1340*
LET2 =$DA63 121.1352* 121.1602
LETREAL =$DA77 121.1352 121.1362*
LETSTR =$DA7A 121.1350 121.1363*
LFTRT =$F465 124.337* 124.398 124.497
LG2 =$E94B 123.195 123.197*
LIN =$D5A8 121.749* 121.799
LINCOOR =$F209 124.52* 124.73 124.81

LINGET	=$DA0C	121.594	121.890	121.899	121.1203	121.1303
		121.1309*	124.186	124.191		
LINKSET	=$D4F2	121.637	121.656*	121.1163	124.213	
LINNUM	=$50	121.79*	121.649	121.649	121.821	121.827
		121.903	121.904	121.907	121.908	121.919
		121.921	121.1149	121.1149	121.1156	121.1156
		121.1165	121.1206	121.1310	121.1311	121.1315
		121.1324	121.1329	121.1330	121.1331	121.1331
		121.1332	121.1333	121.1334	121.1334	121.1336
		122.1353	122.1354	122.1357	122.1359	122.1363
		122.1366	122.1368	122.1374	122.1385	123.1240
		123.1241	123.1243	123.1244	123.1246	123.1247
		123.1250	123.1251	123.1253	123.1254	124.20
		124.130	124.130	124.134	124.137	124.144
		124.144	124.146	124.146	124.148	124.148
		124.194	124.194	124.234	124.235	124.244
		124.247	124.687	124.688	124.793	124.801
		124.819	124.819			
LINPRT	=$ED24	121.925	121.1073	123.727*		
LIST	=$D6A5	121.266	121.884*			
LISTED	=$D724	121.911	121.923	121.947*		
LISTLOOP	=$D6FE	121.927*	121.980			
LOAD	=$D8C9	121.260	121.1154*			
LOCK	=$D6	121.147*	121.710	121.843	121.1157	121.1160
		121.1172	123.1265			
LOG	=$E941	121.280	123.193*	123.938		
LOGSER	=$E918	123.183*	123.211	123.212		
LOGe	=$EEDB	123.951*	123.962	123.963		
LOGtwo	=$E93C	123.191*	123.219	123.220		
LOMEM	=$F2A6	121.242	124.142*			
LORES	=$C056	121.183*	124.215			
LOWSCR	=$C054	121.181*	124.219	124.260		
LOWTR	=$9B	121.114*	121.489	121.492	121.600	121.603
		121.604	121.606	121.614	121.617	121.654
		121.815	121.816	121.817	121.822	121.829
		121.833	121.836	121.910	121.915	121.918
		121.937	121.940	121.943	121.944	121.945
		121.1220	121.1221	122.409	122.410	122.416
		122.420	122.424	122.455	122.455	122.471
		122.474	122.478	122.480	122.483	122.493
		122.494	122.553	122.554	122.560	122.565
		122.568	122.570	122.573	122.574	122.592
		122.608	122.613	122.622	122.634	122.637
		122.669	122.671	122.674	122.675	122.679
		122.690	122.695	122.743	122.746	122.991
		122.991	122.1075	122.1078	122.1080	122.1081
		122.1101	122.1102	124.188	124.188	124.196
		124.196	124.200	124.202	124.202	124.204
		124.204	124.227	124.230	124.243	124.246
		124.819	124.819	124.821		
LR1	=$F46E	124.342*	124.362			
LRUD1	=$F4B3	124.381*	124.614			
LRUD2	=$F4B4	124.382*	124.620			
LRUD3	=$F4C4	124.376	124.380	124.388	124.390*	
LRUD4	=$F4C8	124.372	124.384	124.392*		
LRUDX1	=$F49C	124.369*	124.666			
LRUDX2	=$F49D	124.370*	124.670			

LST1LIN =$D6F7 121.922 121.924*
LSTD? =$D6F5 121.920 121.923*
MAF =$EB66 123.471* 123.970
MAF2 =$EB68 123.472* 123.475
MAININP =$DBEB 121.1518 121.1545*
MAINLST =$D6CC 121.893 121.901*
MAKARY =$E1C1 122.597 122.601*
MAKDIGIT=$EDEE 123.831 123.834*
MAKINT =$E102 122.503* 122.529
MAKSTR =$EDBD 123.799 123.810*
MARKEND =$EE5A 123.867 123.891*
MATHTBL =$D0B2 121.298* 122.63 122.79 122.83 122.87
122.89 122.98 122.168 122.201
MB =$ED57 123.758 123.760*
MEMERR =$D410 121.522 121.526 121.552 121.554 121.557*
122.699 124.132 124.249 124.805
MEMFOUND=$F195 123.1248 123.1253*
MEMSIZ =$73 121.92* 121.638 121.638 121.859 121.859
122.984 122.985 123.1256 123.1257 124.135
124.138 124.144 124.144 124.789 124.795
124.806 124.812
MFA =$EB55 123.461* 123.935
MFA2 =$EB59 123.463* 123.466
MI1 =$E119 122.506 122.513*
MI2 =$E11B 122.509 122.514*
MIDSTR =$E691 121.294 122.1247*
MIN =$DECE 122.153 122.201*
MINUS =$D0C7 121.312* 122.201
MISMTCH =$DD76 121.1717 121.1720*
MIXCLR =$C052 121.179* 124.256
MIXSET =$C053 121.180* 124.216 124.261
MKINT =$E108 122.505* 122.1309
MOK =$F3D2 124.248 124.250*
MONPLOT =$F800 121.196* 124.71
MONREAD =$FEFD 121.203* 121.1155 121.1159 124.241 124.251
124.787
MONREAD2=$FF02 121.204* 124.817
MONWAIT =$FCA8 121.191* 121.1487
MORNAM =$E007 122.361 122.369*
MOV1F =$EB21 123.432* 123.1005 123.1108
MOV2F =$EB1E 123.430* 123.1015
MOVAF =$EB63 123.319 123.336 123.470* 123.689 123.913
123.1077 123.1082
MOVDWN =$F365 124.197 124.199*
MOVESTR =$E5E6 122.1133 122.1152* 122.1238
MOVEX =$F57C 124.496* 124.515
MOVEX2 =$F581 124.495 124.499*
MOVFA =$EB53 123.33 123.460*
MOVFM =$EAF9 121.1028 121.1678 122.224 123.341 123.408*
123.916 123.1049 123.1117
MOVINS =$E5D4 121.1387 122.1129 122.1140*
MOVIT =$F367 124.200* 124.205
MOVMF =$EB2B 122.853 123.434 123.437* 123.925 123.1071
? MOVML =$EB23 123.433*
MOVSTR =$E5E2 122.930 122.1150*
MSKTBL =$F5B2 124.318 124.525*
MSLUP =$EDC1 123.812* 123.827 123.829 123.856

```
   MU1       =$E2B6  122.727      122.747*
   MU2       =$E2C0  122.752*     122.771
   MU3       =$E2D9  122.761      122.770*
   MUL10     =$EA39  123.319*     123.667      123.679      123.775
   MULT      =$E2AD  122.638      122.706      122.742*
   MVBACK    =$EE19  123.858*     123.861
   MVBYT     =$D3C3  121.508*     121.511
   MVDG      =$ECF7  123.700      123.705*
   MVDWN     =$D4A7  121.625      121.628*     121.631      121.635
   MVPRG     =$D4D1  121.645      121.647*
   MVS2      =$E5EA  122.1155*    122.1159
   MVS3      =$E5F3  122.1153     122.1161*
   MVZP      =$F152  123.1220*    123.1224
   NAMOK     =$DFF7  122.356      122.358*
   NB        =$E630  122.1183     122.1185     122.1187     122.1192*
   NCR       =$D712  121.932      121.936*
M  ND        =$DDEE
   NDP       =$ECC8  123.677      123.679*
   NDX       =$F388  124.208      124.210*
   NEEDEX?   =$EE26  123.863      123.865*
MD NEG       =$D000  123.23       123.130      123.131      123.132      123.133
                     123.134      123.135      123.948      123.1096
   NEG2      =$E8A4  123.131*     123.566
   NEGATE    =$DB00  121.1426*    121.1469
   NEGFAC    =$E89E  123.76       123.130*
   NEGNUM    =$E0FE  122.501*     122.510      122.511
   NEGOP     =$EED0  121.313      123.673      123.946*     123.993      123.1097
                     123.1103     123.1145     123.1164
   NEW       =$D649  121.269      121.841*
   NEWDESC   =$DAA1  121.1376     121.1379     121.1383*
   NEWLN?    =$D4B5  121.598      121.636*
   NEWNDX    =$F478  124.348*     124.366      124.368
   NEWSTT    =$D7D2  121.715      121.949      121.1039*    121.1077     121.1201
                     121.1700     124.174      124.180
   NEWVAR    =$E09C  122.442      122.446      122.455*
   NEXT      =$DCF9  121.208      121.1658*
   NEXTCHR   =$DB2F  121.1416     121.1433     121.1437     121.1448     121.1453*
M  NI        =$F6B4
   NIN       =$E036  122.382      122.391*
   NINT      =$E1D5  122.609      122.611*
   NINTA     =$E292  122.719      122.721*
   NL1       =$D49F  121.620      121.623*
   NOI       =$D54C  121.691      121.697*
   NORMAL    =$F273  121.235      123.1209     124.116*
   NOSUCH    =$D647  121.818      121.838*
   NOT?      =$DE90  122.157      122.166*
   NOTFND    =$E087  122.414      122.439*
   NOTFOUND  =$E1B8  122.558      122.596*
   NOTMATH   =$DE35  122.51       122.53       122.118*
   NOTOK     =$D83F  121.1082     121.1092*
   NOTRACE   =$F26F  121.234      124.112*
   NOTZ      =$DE9F  122.172      122.175*
   NOTZE     =$ED4F  123.754      123.756*
   NOTinstr  =$DF3A  122.244      122.261*
   NRM       =$F279  124.117      124.119*
   NTD       =$DCD1  121.1642     121.1644*
   NTL       =$E5B7  122.1124     122.1128*
```

NUM10 =$EA50 123.332* 123.337 123.338
NUMBER =$DE69 122.147* 122.151
NUMCMP =$DFB0 122.294 122.322*
NUMDIM =$0F 121.54* 122.490 122.546 122.590 122.618
122.642 122.680 122.714
NUMIN =$DC63 121.1571 121.1596*
NV1 =$E05B 122.410* 122.426
NV2 =$E065 122.412 122.415*
NWV =$E0B2 122.459 122.461*
NX =$D5A7 121.746 121.748* 121.751
NXARY =$E188 122.563 122.567*
NXBYT =$D3C7 121.505 121.507 121.510* 121.517
NXCHAR =$DB44 121.1462* 121.1468 121.1470
NXCHR =$D56D 121.718* 121.780
NXCLM =$DB0E 121.1431 121.1434*
NXDIG =$DA40 121.1335 121.1337*
NXDIGIT =$EC64 123.614 123.622*
NXDIM =$DFD6 122.343* 122.347
NXDM =$E281 122.705 122.712*
NXFOR =$D38B 121.470 121.474*
NXIN =$DBDC 121.1538* 121.1566
NXINP =$DBF1 121.1548* 121.1613
NXLIN =$D45C 121.590 121.593*
NXLINK =$D4FE 121.659* 121.680
NXLST =$D6DA 121.905 121.909* 121.946
NXNUM =$DA00 121.1299 121.1302*
NXOP =$DDD6 122.66* 122.85
NXPAG =$D3CE 121.496 121.514*
NXPG =$E22E 122.659 122.663*
NXPTR =$E073 122.417 122.423*
NXS =$DCB9 121.1621 121.1633*
NXSFT =$E8DC 123.151* 123.162 123.163
NXSPC =$DB2C 121.1451* 121.1456
NXTDIM =$E12C 122.526* 122.545
NXTERM =$EF96 123.1030 123.1032*
NXVAR =$E059 122.409* 122.428
NXwoFOR =$D260 121.429* 121.1666
NY =$D5A2 121.745* 121.754
NZ =$E409 122.912 122.914*
ODDSER =$EF5C 123.213 123.1003* 123.1106 123.1155
OLDLIN =$77 121.94* 121.1129 121.1129 121.1146 121.1146
OLDTEXT =$79 121.95* 121.875 121.1047 121.1048 121.1127
121.1128 121.1140 121.1143 121.1509 121.1509
124.169 124.169
ONCNT =$D9F8 121.1295 121.1298* 121.1305
ONE =$E913 121.1026 121.1027 123.182* 123.208 123.209
123.1150 123.1151
ONERR =$F2CB 121.243 124.151*
ONESTEP =$D7C3 121.1031 121.1034*
ONGOTO =$D9EC 121.258 121.1292*
OOR =$EF24 123.974* 123.979
OR =$DF4F 121.311 122.272*
ORMASK =$F3 121.165* 121.1482 124.121
OUTDO =$DB5C 121.566 121.929 121.948 121.973 121.976
121.978 121.1070 121.1425 121.1465 121.1479*
OUTOFRNG=$EA2B 123.309* 123.974
OUTPORT =$FE95 121.195* 124.31

OUTQUES =$DB5A 121.563 121.1478* 121.1528 121.1538 121.1565
OUTSP =$DB57 121.1074 121.1455 121.1476*
OVERFLOW=$E8D5 123.122 123.147* 123.315 123.704
OVFLOW =$D2A5 121.434* 123.147
OofDATA =$D28A 121.432* 121.1622 122.598
OofMEM =$D2AD 121.435* 121.557 122.976
PARCHK =$DEB2 122.183 122.186* 122.261 122.836
PARITY? =$EDE5 123.826 123.829*
PARSE =$D56C 121.711 121.717* 121.722
PDL =$DFCD 121.276 122.339*
PECHR =$DC4C 121.1581 121.1586*
PEEK =$E764 121.286 122.1357*
PENCHR =$DC4B 121.1578 121.1585*
PIdoub =$F06B 123.1078 123.1079 123.1130*
PIhalf =$F066 123.1073 123.1074 123.1129* 123.1159 123.1160
PLOT =$F225 121.219 124.66*
PLOTFNS =$F1EC 122.227 124.37* 124.52 124.66
PLU? =$D604 121.794 121.796*
PLUS =$DOCD 121.316* 122.79
PLUSEPS =$E8C6 123.138* 123.486
PNTR =$OF 121.53* 121.596 121.641 121.651 121.741
121.757 121.791
PNTSTK =$EE5F 123.889 123.893*
POINT =$E73D 121.1593 122.164 122.1339*
POKE =$E77B 121.263 122.1371*
POP =$D96B 121.239 121.255 121.1224*
POS =$E2FF 121.277 122.791*
POSOP =$DF65 121.317 122.284*
PREAD =$FB1E 121.193* 122.340
PREFNC =$DDF6 122.70 122.83*
PREFTEST=$DDCD 122.62* 122.81
PRGEND =$AF 121.136* 121.593 121.593 121.662 121.662
121.851 121.855 121.1149 121.1149 124.185
124.185
PRINT =$DAD5 121.264 121.1405*
PRINT2 =$DAD7 121.1406* 121.1454
PRNTFAC =$ED2E 123.732*
PRNTIN? =$D431 121.573* 121.1135
PRNU =$F1E5 121.216 124.29*
PROGIO =$D901 121.1152 121.1158 121.1175*
PROMPT =$33 121.74* 121.683
PRSTR =$ED31 123.724 123.733*
PRSTRING=$DACF 121.1403* 121.1419 121.1422
PRTOK =$D750 121.968 121.972*
PSHF =$DE10 122.91 122.97*
PSHFACX =$DE15 121.1035 122.103*
PSHMAD =$DDFD 122.67 122.87*
PTRGET =$DFE3 121.1340 121.1548 121.1661 122.206 122.350*
122.809 124.830
PTRGET2 =$DFE8 122.345 122.352*
PTRGET3 =$DFEA 122.353* 122.829
PULL =$DOOO 121.1245 122.132 122.133 122.134 122.530
122.548 122.838 122.862 122.867 122.1119
122.1281
PULL3 =$D9C5 121.1241 121.1271*
PUSH =$DOOO 121.1012 121.1036 121.1195 121.1196 121.1345
122.112 122.113 122.251 122.528 122.815

		122.816	122.835	122.854	122.860	122.1116
PUSHFAC	=$DE20	121.1025	122.111*			
PUTCHR	=$DC3F	121.1573	121.1579*			
PUTDP	=$EDAA	123.798	123.800*			
PUTEMP	=$E435	122.933	122.937*			
PUTEX	=$EE36	123.868	123.874*			
PUTIN	=$D5CD	121.726	121.730	121.734	121.767*	121.783
		121.785				
PUTLINK	=$D50F	121.661	121.664*			
PUTNEW	=$E42A	122.925	122.931*	122.1137	122.1216	122.1239
PUTSTR	=$DA7B	121.1364*	121.1594			
PUTTOK	=$D5CB	121.759	121.766*	121.801		
PW1	=$EEA0	123.920	123.923*			
PW2	=$EEBA	123.927	123.932	123.935*		
QDRNT	=$D3	121.144*	124.393	124.463	124.470	124.487
		124.514	124.589	124.641		
QI1	=$EC06	123.562	123.568*			
QI2	=$EC12	123.570	123.575*			
QINT	=$EBF2	122.514	122.1350	123.557*	123.588	123.782
QOUT	=$DBC4	121.1522	121.1528*			
QUARTER	=$F070	123.1087	123.1088	123.1098	123.1099	123.1131*
QUO?	=$E404	122.908	122.911*			
RAR	=$E1DE	122.614	122.617*			
RARY	=$E298	122.722	122.725*			
RD1	=$EFCC	123.1046	123.1058*			
RD2	=$EFE7	123.1071*	123.1114			
RD2BIT	=$FCFA	121.186*	124.815			
RDKEY	=$FD0C	121.188*	121.702	121.1557		
READ	=$DBE2	121.213	121.1540*			
READERR	=$DB7B	121.1493	121.1496*			
REASON	=$D3E3	121.481	121.529*	122.602	122.653	123.1271
RECALL	=$F3BC	121.245	124.240*			
REENT	=$DCEF	121.1506	121.1507	121.1655*		
REM	=$D9DC	121.256	121.1284*			
REM?	=$D5E2	121.776	121.778*			
REMN	=$D9A6	121.1204	121.1254*	121.1284	124.157	
REMSTK	=$F8	121.168*	121.1040	124.163		
RES>FAC	=$EAE6	123.241	123.398	123.401*		
RESPERR	=$DB87	121.1492	121.1502*			
RESTART	=$D43C	121.576	121.579*	121.587	121.663	121.1136
		122.366	123.1281	123.1282		
RESTORE	=$D849	121.252	121.863	121.1097*		
RESULT	=$62	121.85*	122.725	122.744	122.747	122.764
		122.767	123.150	123.227	123.228	123.229
		123.230	123.251	123.251	123.252	123.252
		123.253	123.253	123.254	123.254	123.255
		123.256	123.257	123.258	123.371	123.401
		123.401	123.402	123.402		
RESUME	=$F318	121.244	124.176*			
RET1	=$D392	121.462	121.473	121.479*		
RET10	=$DB02	121.1406	121.1427*	121.1463		
RET11	=$DCDE	121.1646	121.1650*			
RET12	=$DD73	121.1718*	121.1719			
RET2	=$D40F	121.530	121.533	121.551	121.555*	
RET3	=$D648	121.823	121.830	121.831	121.839*	121.841
		121.897				
RET4	=$D696	121.858	121.877*	121.889	121.900	

RET5	=$D857	121.1080	121.1105*			
RET6	=$D8AF	121.1121	121.1138	121.1147*		
RET7	=$D96A	121.1222*	121.1224			
RET8	=$D9A2	121.1250*	121.1263	121.1265		
RET9	=$DA0B	121.1307*	121.1312			
RETURN	=$D984	121.1230	121.1238*			
REdimARR	=$D2D8	121.438*	122.581			
RIGHT	=$F48A	124.337	124.359*			
RIGHTSTR	=$E686	121.293	122.1241*			
RM1	=$D9AE	121.1258*	121.1269			
RM2	=$D9B6	121.1262*	121.1268			
RND	=$EFAE	121.279	123.1044*			
RNDADJ1	=$EFA6	123.1041*	123.1052	123.1053		
RNDADJ2	=$EFAA	123.1042*	123.1055	123.1056		
RNDB	=$EB72	121.1353	122.111	123.346	123.437	123.470
		123.482*				
RNDSEED	=$C9	121.140*	123.1047	123.1069		
ROT	=$F721	121.230	124.738*			
ROTZ	=$F9	121.169*	124.739	124.772		
ROUN	=$ED86	123.774	123.781*			
ROUND	=$EB7A	123.486*	123.968			
RS1	=$D3ED	121.531	121.534*			
RS2	=$D3F1	121.537*	121.540			
RS3	=$D3FC	121.543*	121.546			
RTN1	=$E086	122.433	122.437*			
RTN10	=$EB8F	123.491	123.495	123.497*		
RTN11	=$EC11	123.549	123.573*			
RTN12	=$EC49	123.587	123.605*			
RTN13	=$EEDA	123.945	123.947	123.949*		
RTN14	=$EFA5	123.1039*	123.1051			
RTN15	=$F0CD	123.1163	123.1165*			
RTN16	=$F122	123.1187	123.1194*			
RTN2	=$E2AC	122.677	122.736*	122.798		
RTN3	=$E645	122.1197	122.1199	122.1204*		
RTN4	=$E79F	122.1389*	123.39			
RTN5	=$E89D	123.120	123.128*			
RTN6	=$E8D4	123.137	123.139	123.141	123.143	123.145*
RTN7	=$E9E2	123.223	123.263*			
RTN8	=$EA4F	123.321	123.330*			
RTN9	=$EB71	123.477*	123.483	123.485	123.487	
RTNwoGSB	=$D276	121.431*	121.1231			
RTS1	=$F489	124.353	124.357*			
RTS2	=$F600	124.513	124.579*			
RTS3	=$F6F5	124.718*	124.728			
RUN	=$D912	121.250	121.1179*			
RUNLINE	=$D91B	121.1182	121.1184*			
SAMEFOR?	=$D37F	121.464	121.469*			
SAVDIM	=$E1E7	122.623*	122.643			
SAVE	=$D8B0	121.261	121.1149*			
SAVOLD	=$E378	122.848*	122.851			
SAVOP	=$DDD7	122.67*	122.204			
SAVY	=$EE09	123.845	123.849*			
MD SB	=$D000	121.489	121.504	121.1149	121.1149	121.1220
		121.1221	123.71	123.72	123.73	123.74
		123.385	123.386	123.387	123.388	
SCALE	=$F727	121.231	124.742*			
SCALEZ	=$E7	121.159*	124.608	124.660	124.743	

SCDCH =$E03D 122.384 122.395*
SCREEN =$DEF9 122.226* 122.238
SCRN =$F871 121.192* 122.230
SCRTCH =$D64B 121.842* 123.1272
SE =$D578 121.720 121.723*
SEND =$DB64 121.1481 121.1483*
SENDCHR =$D702 121.929* 121.957
SERIES =$EF72 123.996 123.1013*
SERLEN =$A3 121.126* 123.617 123.670 123.1017 123.1037
SERLOOP =$EF89 123.1025* 123.1038
SERMAIN =$EF76 123.1008 123.1015*
SERPNT =$AD 121.133* 123.1003 123.1004 123.1013 123.1014
123.1016 123.1018 123.1022 123.1023 123.1024
123.1026 123.1027 123.1032 123.1033
SETCOL =$F864 121.199* 124.91
SETDA =$D853 121.1101 121.1103* 121.1643
SETDP =$EC98 123.624 123.648*
SETEND =$D3B7 121.501 121.504*
SETFOR =$EB27 121.1362 121.1685 123.435*
SETGR =$FB40 121.201* 124.217
SETHI =$F299 124.131 124.134*
SETHPG =$F3EA 124.258 124.262*
SETPTRS =$D665 121.656 121.714 121.856* 121.1162 121.1183
SETSGN =$EC85 123.630 123.632 123.639*
SETTXT =$FB39 121.202* 124.220
SETVPNT =$E0DE 122.421 122.480*
SFS =$DFA5 122.310 122.312 122.315*
SFSG =$ED41 123.746 123.749*
SGN =$EB90 121.270 123.499*
SGN? =$DEAB 122.179 122.182*
SGNCHK =$EC8C 123.637 123.642*
SGNCPR =$AB 121.131* 122.136 123.25 123.61 123.282
123.305 123.326 123.340 123.694 123.998
123.1085
SGNIF =$E829 123.75* 123.513 123.598
SH1 =$E8FD 123.168* 123.178
SH2 =$E903 123.169 123.171*
SH3 =$E911 123.167 123.179*
SHAPEH =$1B 121.64* 124.268 124.276 124.277 124.584
124.636 124.755 124.764
SHAPEL =$1A 121.63* 124.270 124.272 124.583 124.603
124.628 124.628 124.629 124.635 124.655
124.678 124.678 124.679 124.747 124.747
124.750 124.758 124.759 124.762 124.765
SHAPEPNT=$E8 121.160* 124.747 124.747 124.763 124.809
124.811
SHFTIN =$D5E9 121.782* 121.789
SHFTR =$E907 123.60 123.173* 123.581
SHFTRES =$E8DA 123.150* 123.244
SHIFT =$E8F0 123.29 123.161* 123.571
SHIN =$D5F2 121.725 121.786*
SHLOAD =$F775 121.232 124.780*
SI =$F27B 124.120* 124.126
SI1 =$F023 123.1092 123.1097* 123.1127
SI2 =$F026 123.1095 123.1098*
SI3 =$F033 123.1102 123.1104*
SIGN =$EB82 121.1034 123.193 123.490* 123.499 123.526

		123.1044				
SIGN1	=$EB86	123.492*	123.529			
SIGN2	=$EB88	123.493*	123.553			
SIGNFLG	=$16	121.62*	123.1096	123.1096	123.1110	123.1120
SIGNIF	=$E82E	123.75	123.77*	123.403	123.1068	
SIN	=$EFF1	121.283	123.1077*	123.1111		
SINSER	=$F075	123.1104	123.1105	123.1132*		
SK2	=$D5FD	121.792*	121.797			
SKIP	=$DE64	122.145*	122.155			
SKIPTOK	=$D5F9	121.756	121.762	121.764	121.790*	
SKP	=$DC57	121.1590	121.1592*			
SKPTK	=$D746	121.967*	121.971			
SKPV	=$DD02	121.1660	121.1662*			
SL1	=$F796	124.798	124.800*			
SL2	=$F7A0	124.803	124.805*			
SL3	=$F7A3	124.804	124.806*			
SLI	=$F3AF	124.232	124.234*			
SNDQ?	=$DC1F	121.1556	121.1564*			
SNGFLT	=$E301	122.176	122.232	122.282	122.341	122.792*
		122.1292	122.1304	122.1369		
SNTXERR	=$DE0D	122.45	122.95*			
SPC?	=$DB21	121.1441	121.1443*			
SPEED	=$F262	121.247	124.102*			
SPEEDZ	=$F1	121.163*	121.1486	123.1222	124.107	
SQR	=$EE8D	121.278	123.913*			
SQRhalf	=$E92D	123.188*	123.202	123.203		
SQRtwo	=$E932	123.189*	123.205	123.206		
SS	=$EF85	123.1021	123.1023*			
SSF	=$D5E0	121.774	121.777*			
ST1	=$DE8A	122.161	122.163*			
STACK	=$0100	121.174*	121.460	121.465	121.467	121.469
		121.472	121.1680	121.1690	121.1692	121.1694
		121.1696	121.1698	122.444	122.534	122.536
		122.539	122.541	123.748	123.803	123.808
		123.843	123.848	123.858	123.874	123.876
		123.884	123.886	123.888	123.890	123.892
		123.893	123.894			
STE	=$ED60	123.759	123.764*			
STEP	=$D7AF	121.1021	121.1022	121.1026*		
STEX	=$ED05	123.703	123.715*			
STKINI	=$D683	121.570	121.864*			
STOP	=$D86E	121.257	121.1118*			
STORE	=$F39F	121.246	124.225*			
STR	=$E3C5	121.288	122.878*			
STRCMP	=$DF7D	122.285	122.295*			
STREND	=$6D	121.89*	121.482	121.483	121.650	121.650
		121.861	121.862	122.456	122.456	122.555
		122.557	122.654	122.655	122.668	122.672
		122.780	122.783	122.963	122.966	122.991
		122.991	122.1017	122.1019	124.130	124.130
		124.802				
STRINI	=$E3D5	121.1385	122.889*	122.927	122.1128	
STRIP	=$D541	121.692*	121.696			
STRLIT	=$E3E7	121.1421	121.1457	122.163	122.885	122.896*
STRLT2	=$E3ED	121.1592	122.899*			
STRNG1	=$AB	121.134*	121.1386	121.1386	122.212	122.899
		122.900	122.905	122.916	122.918	122.922

		122.928	122.929	122.1119	122.1119	122.1121
		122.1134	122.1135	122.1141	122.1144	122.1147
	STRNG2 =$AD	121.135*	121.735	121.766	122.605	122.617
		122.639	122.640	122.657	122.658	122.664
		122.682	122.683	122.702	122.703	122.713
		122.716	122.759	122.760	122.917	122.921
		122.1318	122.1318	122.1339	122.1339	123.750
		123.800	123.809	123.839	123.849	123.857
	STRNG? =$E01C	122.372	122.378*			
	STROUT =$DB3A	121.573	121.1457*	121.1508	121.1649	123.733
		123.1277	123.1278			
	STRPRT =$DB3D	121.1403	121.1458*	121.1526		
	STRSPA =$E3DD	122.890*	122.1210	122.1228		
	STRTRNG =$D6B1	121.884	121.885	121.887	121.890*	
	STRTXT =$DE81	121.1523	122.158*			
	STRtoLNG=$D310	121.442*	122.1125			
	STXP =$DC27	121.1562	121.1567*			
	STXTPT =$D697	121.856	121.879*			
MD	SUB =$D000	121.1149				
	SUBERR =$E196	122.576*	122.593	122.697		
	SUBFLG =$14	121.59*	121.876	121.992	122.385	122.397
		122.401	122.405	122.518	122.584	122.596
		122.808	122.828	124.829	124.832	
	SUBMAN =$E806	123.65	123.67*			
	SVAR =$E4B5	122.1004*	122.1009			
	SVARGD =$E4BD	122.1005	122.1008*			
	SVARS =$E4A9	122.997	122.1001*			
	SVP =$E0E8	122.484	122.486*			
	SVY =$EDBB	123.805	123.809*			
	SWPNT =$DC7E	121.1604	121.1606	121.1608*		
	SYNCHR =$DEC0	121.1014	121.1281	121.1344	121.1525	122.193*
		122.813	122.826	124.42	124.60	124.152
		124.190	124.699	124.729	124.769	
	SYNERR =$DEC9	121.1095	121.1236	121.1442	121.1500	122.95
		122.195	122.198*	122.357	122.865	124.182
	SYNTXERR=$D270	121.430*	122.198			
	TAB =$DB03	121.1414	121.1429*			
	TABIT =$DB2B	121.1444	121.1450*			
	TABV =$FB5B	121.200*	124.100			
	TABWHERE=$DB16	121.1408	121.1411	121.1438*		
	TAN =$F03A	121.284	123.1108*			
	TAN2 =$F062	123.1121	123.1126*			
	TAPEPNT =$F7BC	124.237	124.250	124.819*		
	TEMP1 =$93	121.118*	121.535	121.538	121.542	123.432
		123.1006	123.1009	123.1115		
	TEMP2 =$98	121.119*	123.430	123.1035		
	TEMP3 =$8A	121.120*	123.923	123.929	123.939	123.1112
		123.1122				
	TEMPPT =$52	121.80*	121.865	121.1157	121.1169	122.931
		122.952	122.996	122.1200	123.1237	
	TEMPST =$55	121.82*	121.864	122.932	122.992	122.993
	TEND =$EE8D	123.855	123.911*			
	TEXT =$F399	121.215	124.219*			
	TMPEXP =$99	121.121*	123.611	123.659	123.678	123.764
		123.776	123.779	123.784	123.796	123.844
	TOK? =$D588	121.728	121.731*			
	TOKDONE =$D75F	121.975	121.978*			

TOKEN? =$D734 121.938 121.957*
TOKLP =$D749 121.969* 121.970
TOKLUP =$D755 121.974* 121.977
TOKTABL =$D0D0 121.319* 121.736 121.738 121.962 121.964
TR =$D000 121.602 121.603 121.662 121.662 121.1157
122.1083 123.401 123.401 123.402 123.402
124.148 124.148 124.153 124.153 124.156
124.156 124.168 124.168 124.169 124.169
124.170 124.170 124.171 124.171 124.176
124.176 124.177 124.177 124.188 124.188
124.747 124.747 124.823 124.823
TR1 =$D81D 121.1065 121.1068 121.1075*
TRACE =$F26D 121.233 124.110*
TRACE? =$D805 121.592 121.1064* 121.1094
MD TRAX =$D000 122.991 122.1003 122.1082 122.1103
MD TRAY =$D000 121.638 121.649 121.650 121.657 121.859
121.860 121.1129 121.1146 121.1175 121.1176
121.1386 121.1509 121.1551 121.1608 121.1609
122.455 122.456
TRCFLG =$F2 121.164* 121.1064 123.1225 124.113
MD TRDB =$D000 121.662 123.401 123.402 124.148 124.153
124.156 124.168 124.169 124.170 124.171
124.176 124.177 124.188 124.747 124.823
TRUE =$DF60 122.274 122.278 122.281*
TRUE? =$D9D8 121.1279 121.1282*
MD TRX =$D000 121.593 124.185
MD TRXY =$D000 121.1552 122.889 122.1318 122.1339
TVAR =$E4A0 122.996* 122.999
TWOBRK =$E09A 122.447 122.448 122.450*
TXPSV =$87 121.102* 121.1551 121.1551 121.1609 121.1609
TXTCLR =$C050 121.178* 124.264
TXTPSV =$F4 121.166* 124.153 124.153 124.170 124.170
TXTPTR =$B8 121.139* 121.582 121.583 121.706 121.743
121.790 121.803 121.805 121.880 121.881
121.1007 121.1009 121.1042 121.1043 121.1050
121.1053 121.1057 121.1060 121.1063 121.1063
121.1063 121.1122 121.1123 121.1144 121.1145
121.1195 121.1195 121.1210 121.1211 121.1220
121.1221 121.1245 121.1245 121.1249 121.1249
121.1249 121.1262 121.1509 121.1509 121.1551
121.1551 121.1552 121.1552 121.1567 121.1568
121.1575 121.1587 121.1588 121.1608 121.1608
121.1609 121.1609 121.1624 121.1627 121.1630
121.1633 121.1697 121.1699 122.23 122.23
122.23 122.78 122.78 122.78 122.158
122.159 122.194 122.816 122.816 122.854
122.854 122.856 122.859 122.867 122.867
122.1318 122.1318 122.1320 122.1325 122.1339
122.1339 123.712 123.1184 123.1184 124.153
124.153 124.170 124.170 124.177 124.177
TXTTAB =$67 121.86* 121.657 121.657 121.812 121.813
121.845 121.847 121.848 121.852 121.880
121.881 121.1098 121.1100 121.1149 121.1149
121.1156 121.1156 121.1175 121.1175 121.1215
121.1216 123.1262 123.1263 123.1267 123.1268
123.1268 123.1269 123.1270
TYPEMISS=$D303 121.441* 121.1720

UNARY =$DF0C 122.184 122.237*
UNDERR =$D97C 121.1218 121.1233*
UNDFNC =$E30E 122.801* 122.845
UNDFUNC =$D340 121.445* 122.801
UNDSTAT =$D2BA 121.436* 121.1233
UNFNC =$D080 121.270* 122.264 122.266
? UP =$F4D5 124.400*
UP1 =$F4EB 124.407 124.411*
UP2 =$F4FB 124.405 124.420*
UP3 =$F4FD 124.410 124.419 124.421*
UP4 =$F4FF 124.403 124.422*
UP5 =$F4F6 124.415 124.417*
UPDOWN =$F4D3 124.399* 124.516
UPDWN1 =$F501 124.423* 124.430 124.447
USR =$0A 121.50* 121.273 123.1214 123.1217 123.1218
V2 =$2D 121.71* 124.47 124.57
VAL =$E707 121.289 122.1315*
VALTYP =$11 121.56* 121.1345 121.1345 121.1418 121.1570
121.1601 121.1715 122.54 122.75 122.144
122.209 122.214 122.296 122.359 122.360
122.381 122.388 122.521 122.523 122.548
122.548 122.774 122.785 122.947 122.1295
VAR? =$DE6C 122.146 122.148*
VARL =$DED5 122.149 122.206* 122.441 122.445
VARNAM =$81 121.99* 122.353 122.389 122.390 122.395
122.415 122.418 122.470 122.473 122.528
122.528 122.530 122.530 122.562 122.564
122.607 122.612 122.718 122.721
VARNXT =$DCFF 121.1658 121.1661* 121.1709
VARPNT =$83 121.100* 122.486 122.487 122.730 122.733
122.735 122.815 122.815 122.841 122.846
122.848 122.852 122.860 122.860 123.838
123.850
VARTAB =$69 121.87* 121.593 121.593 121.602 121.608
121.609 121.611 121.613 121.618 121.639
121.643 121.650 121.650 121.662 121.662
121.850 121.854 121.860 121.860 121.1156
121.1156 121.1176 121.1176 121.1374 121.1378
122.406 122.407 122.1003 122.1003 124.146
124.146 124.148 124.148 124.185 124.185
124.204 124.204 124.211 124.212
VARTIO =$D8F0 121.1150 121.1154 121.1165* 124.833
VDONE =$E55D 122.1087* 122.1095
VL2 =$E70F 122.1316 122.1318*
VL3 =$E727 122.1326 122.1328*
VLIN =$F241 121.221 124.81*
VLINE =$F828 121.198* 124.87
VPNT =$A0 121.116* 121.1356 121.1359 121.1365 121.1370
121.1373 121.1377 121.1380 121.1381 121.1384
122.207 122.208 122.217 122.220 122.251
122.251
VR1 =$DEE5 122.210 122.214*
VR2 =$DEF6 122.215 122.224*
VSEARCH =$E04F 122.402 122.404*
VTAB =$F256 121.240 124.95*
WAIT =$E784 121.259 122.1377*
WHATX =$EE42 123.880* 123.882

```
   WNDUP    =$EE57  123.755     123.890*
   WNX      =$DC72  121.1595    121.1603*
   WRITE    =$FECD  121.189*    121.1151    121.1153    124.236     124.238
   WT2      =$E793  122.1381    122.1383*
   WT3      =$E797  122.1385*   122.1388
   XOH      =$E1    121.154*    124.286     124.454     124.462     124.478
                    124.572     124.577
   XOL      =$E0    121.153*    124.285     124.453     124.459     124.477
                    124.578
   X1       =$EF19  123.967     123.969*
   X2       =$EF27  123.973     123.975*
   X3       =$EF37  123.984*    123.989
   XDRAW    =$F76F  121.227     124.777*
?  XDRAW0   =$F65D  124.635*
   XDRAW1   =$F661  124.637*    124.778
   XDRAW2   =$F682  124.656*    124.677     124.680
   XDRAW3   =$F68C  124.661*    124.672
   XDRAW4   =$F699  124.665     124.668*
   XDRAW5   =$F6A4  124.669     124.671*
   YO       =$E2    121.155*    124.284     124.455     124.481     124.486
                    124.554     124.559     124.560
   ZARY     =$E21A  122.650     122.653*
   ZEND     =$E408  122.906     122.913*
   ZERO     =$EA31  123.293     123.300     123.312*
   ZEROFAC  =$E84E  122.1317    123.94*     123.116     123.314
   ZF       =$DBE9  121.1535    121.1544*
   ZFAC     =$EC40  123.558     123.600*
   ZLUP     =$E229  122.660*    122.662     122.665
   ZPSTUFF  =$F10B  123.1184*   123.1189    123.1220
   at       =$C5    121.36*     121.758     121.765     124.59      124.767
   data     =$83    121.25*     121.775     121.1636
   equal    =$D0    121.41*     121.1343    122.812
   fn       =$C2    121.33*     122.178     122.825
   for      =$81    121.24*     121.461     121.1037
   gosub    =$B0    121.28*     121.1197    121.1229    121.1294
   goto     =$AB    121.27*     121.1278    121.1296    124.151
   leftstr  =$E8    121.44*     122.243
   minus    =$C9    121.40*     121.886     121.894     122.152     123.629
   not      =$C6    121.37*     122.166
   plus     =$C8    121.39*     122.52      122.154     123.633
   pop      =$A1    121.26*     121.1240
   print    =$BA    121.30*     121.729
   rem      =$B2    121.29*     121.779
   scrn     =$D7    121.43*     122.237
   sgn      =$D2    121.42*     122.182
   spc      =$C3    121.34*     121.1409
   step     =$C7    121.38*     121.1030
   tab      =$C0    121.31*     121.1407
   then     =$C4    121.35*     121.1280
   to       =$C1    121.32*     121.1013    124.722     124.727
```

Cross referenced symbol table - numerical order:

```
   GOWARM   =$00    121.48*     123.1205    123.1206    123.1211    123.1283
                    123.1284    123.1285
   GOSTROUT=$03     121.49*     123.1207    123.1208    123.1212    123.1279
```

		123.1280				
USR	=$0A	121.50*	121.273	123.1214	123.1217	123.1218
CHARAC	=$0D	121.51*	121.1255	121.1259	121.1260	121.1314
		121.1334	121.1577	121.1579	121.1583	122.897
		122.907	123.597	123.934	123.976	
ENDCHR	=$0E	121.52*	121.723	121.781	121.784	121.1257
		121.1258	121.1261	121.1264	121.1586	122.898
		122.909				
PNTR	=$0F	121.53*	121.596	121.641	121.651	121.741
		121.757	121.791			
NUMDIM	=$0F	121.54*	122.490	122.546	122.590	122.618
		122.642	122.680	122.714		
DIMFLG	=$10	121.55*	122.352	122.520	122.550	122.582
		122.625	122.676			
VALTYP	=$11	121.56*	121.1345	121.1345	121.1418	121.1570
		121.1601	121.1715	122.54	122.75	122.144
		122.209	122.214	122.296	122.359	122.360
		122.381	122.388	122.521	122.523	122.548
		122.548	122.774	122.785	122.947	122.1295
DATAFLG	=$13	121.57*	121.58	121.709	121.719	121.777
GARFLG	=$13	121.58*	122.955	122.977	122.981	
SUBFLG	=$14	121.59*	121.876	121.992	122.385	122.397
		122.401	122.405	122.518	122.584	122.596
		122.808	122.828	124.829	124.832	
INPUTFLG	=$15	121.60*	121.1491	121.1545	121.1555	121.1572
		121.1614	121.1641			
CPRMASK	=$16	121.61*	122.131	122.334		
SIGNFLG	=$16	121.62*	123.1096	123.1096	123.1110	123.1120
SHAPEL	=$1A	121.63*	124.270	124.272	124.583	124.603
		124.628	124.628	124.629	124.635	124.655
		124.678	124.678	124.679	124.747	124.747
		124.750	124.758	124.759	124.762	124.765
SHAPEH	=$1B	121.64*	124.268	124.276	124.277	124.584
		124.636	124.755	124.764		
HCOLOR1	=$1C	121.65*	124.266	124.271	124.323	124.328
		124.350	124.354	124.356	124.386	124.506
COUNTH	=$1D	121.66*	124.493	124.512		
CH	=$24	121.67*	121.930	121.935	121.1429	121.1436
		121.1447	122.791	124.848		
GBASL	=$26	121.68*	124.289	124.292	124.293	124.303
		124.329	124.331	124.332	124.375	124.385
		124.390	124.391	124.404	124.413	124.417
		124.421	124.431	124.437	124.442	124.446
		124.505	124.508	124.509	124.545	124.550
GBASH	=$27	121.69*	124.295	124.299	124.301	124.304
		124.307	124.401	124.423	124.426	124.547
		124.555				
H2	=$2C	121.70*	124.46	124.56	124.75	
V2	=$2D	121.71*	124.47	124.57		
HMASK	=$30	121.72*	124.319	124.330	124.338	124.342
		124.348	124.359	124.374	124.379	124.387
		124.507	124.567			
INVFLG	=$32	121.73*	124.120			
PROMPT	=$33	121.74*	121.683			
A1L	=$3C	121.75*	121.1167	121.1175	121.1175	124.784
		124.814	124.823	124.823		
A1H	=$3D	121.76*	121.1168	124.781	124.808	

A2L	=$3E	121.77*	121.1170	121.1176	121.1176	124.786
		124.792	124.819	124.819		
A2H	=$3F	121.78*	121.1171	124.782	124.800	
LINNUM	=$50	121.79*	121.649	121.649	121.821	121.827
		121.903	121.904	121.907	121.908	121.919
		121.921	121.1149	121.1149	121.1156	121.1156
		121.1165	121.1206	121.1310	121.1311	121.1315
		121.1324	121.1329	121.1330	121.1331	121.1331
		121.1332	121.1333	121.1334	121.1334	121.1336
		122.1353	122.1354	122.1357	122.1359	122.1363
		122.1366	122.1368	122.1374	122.1385	123.1240
		123.1241	123.1243	123.1244	123.1246	123.1247
		123.1250	123.1251	123.1253	123.1254	124.20
		124.130	124.130	124.134	124.137	124.144
		124.144	124.146	124.146	124.148	124.148
		124.194	124.194	124.234	124.235	124.244
		124.247	124.687	124.688	124.793	124.801
		124.819	124.819			
TEMPPT	=$52	121.80*	121.865	121.1157	121.1169	122.931
		122.952	122.996	122.1200	123.1237	
LASTPT	=$53	121.81*	122.948	122.1196	122.1198	122.1202
		123.1228				
TEMPST	=$55	121.82*	121.864	122.932	122.992	122.993
INDEX	=$5E	121.83*	121.489	121.499	121.504	121.523
		121.525	121.601	121.602	121.624	121.626
		121.628	121.632	121.657	121.657	121.660
		121.666	121.670	121.673	121.674	121.677
		121.678	121.679	121.1023	121.1024	121.1316
		121.1327	121.1331	121.1464	122.58	122.60
		122.105	122.106	122.108	122.116	122.641
		122.711	122.742	122.994	122.995	122.1003
		122.1003	122.1023	122.1024	122.1026	122.1029
		122.1032	122.1036	122.1044	122.1048	122.1048
		122.1048	122.1049	122.1057	122.1060	122.1063
		122.1066	122.1069	122.1082	122.1082	122.1086
		122.1086	122.1086	122.1087	122.1150	122.1151
		122.1156	122.1168	122.1169	122.1173	122.1176
		122.1179	122.1192	122.1193	122.1236	122.1236
		122.1236	122.1302	122.1319	122.1322	122.1324
		123.267	123.268	123.270	123.273	123.276
		123.279	123.287	123.408	123.409	123.411
		123.414	123.417	123.420	123.425	123.438
		123.439	123.442	123.445	123.448	123.453
		123.456				
DEST	=$60	121.84*	121.603	121.610	121.622	121.629
		121.633	121.1676	122.1323	122.1328	122.1330
		122.1333	122.1338	123.520	123.521	123.523
		123.527	123.532	123.537	123.541	123.547
		124.188	124.188	124.196	124.196	124.201
		124.203	124.203	124.206	124.207	
RESULT	=$62	121.85*	122.725	122.744	122.747	122.764
		122.767	123.150	123.227	123.228	123.229
		123.230	123.251	123.251	123.252	123.252
		123.253	123.253	123.254	123.254	123.255
		123.256	123.257	123.258	123.371	123.401
		123.401	123.402	123.402		
TXTTAB	=$67	121.86*	121.657	121.657	121.812	121.813

		121.845	121.847	121.848	121.852	121.880
		121.881	121.1098	121.1100	121.1149	121.1149
		121.1156	121.1156	121.1175	121.1175	121.1215
		121.1216	123.1262	123.1263	123.1267	123.1268
		123.1268	123.1269	123.1270		
VARTAB	=$69	121.87*	121.593	121.593	121.602	121.608
		121.609	121.611	121.613	121.618	121.639
		121.643	121.650	121.650	121.662	121.662
		121.850	121.854	121.860	121.860	121.1156
		121.1156	121.1176	121.1176	121.1374	121.1378
		122.406	122.407	122.1003	122.1003	124.146
		124.146	124.148	124.148	124.185	124.185
		124.204	124.204	124.211	124.212	
ARYTAB	=$6B	121.88*	121.860	121.860	122.411	122.413
		122.455	122.455	122.467	122.468	122.551
		122.552	122.1004	122.1006		
STREND	=$6D	121.89*	121.482	121.483	121.650	121.650
		121.861	121.862	122.456	122.456	122.555
		122.557	122.654	122.655	122.668	122.672
		122.780	122.783	122.963	122.966	122.991
		122.991	122.1017	122.1019	124.130	124.130
		124.802				
FRETOP	=$6F	121.90*	121.529	121.532	121.550	121.553
		121.638	121.638	121.859	121.859	121.1366
		121.1371	122.779	122.782	122.959	122.960
		122.968	122.969	122.986	122.987	122.1070
		122.1073	122.1103	122.1103	122.1184	122.1186
		122.1190	122.1190	122.1190	123.1258	123.1259
		124.136	124.139	124.807	124.813	
FRESPC	=$71	121.91*	122.970	122.971	122.1157	122.1162
		122.1162	122.1162			
MEMSIZ	=$73	121.92*	121.638	121.638	121.859	121.859
		122.984	122.985	123.1256	123.1257	124.135
		124.138	124.144	124.144	124.789	124.795
		124.806	124.812			
CURLIN	=$75	121.93*	121.574	121.589	121.1012	121.1012
		121.1044	121.1058	121.1061	121.1066	121.1071
		121.1072	121.1124	121.1129	121.1129	121.1146
		121.1146	121.1180	121.1196	121.1196	121.1205
		121.1242	121.1244	121.1498	121.1499	121.1693
		121.1695	122.796	123.725	123.726	123.1199
		124.156	124.156	124.168	124.168	124.171
		124.171	124.176	124.176		
OLDLIN	=$77	121.94*	121.1129	121.1129	121.1146	121.1146
OLDTEXT	=$79	121.95*	121.875	121.1047	121.1048	121.1127
		121.1128	121.1140	121.1143	121.1509	121.1509
		124.169	124.169			
DATLIN	=$7B	121.96*	121.1496	121.1497	121.1628	121.1632
DATPTR	=$7D	121.97*	121.1103	121.1104	121.1540	121.1541
INPTR	=$7F	121.98*	121.1546	121.1547	121.1552	121.1552
		121.1608	121.1608	121.1639	121.1640	121.1645
for	=$81	121.24*	121.461	121.1037		
VARNAM	=$81	121.99*	122.353	122.389	122.390	122.395
		122.415	122.418	122.470	122.473	122.528
		122.528	122.530	122.530	122.562	122.564
		122.607	122.612	122.718	122.721	
data	=$83	121.25*	121.775	121.1636		

VARPNT	=$83	121.100*	122.486	122.487	122.730	122.733
		122.735	122.815	122.815	122.841	122.846
		122.848	122.852	122.860	122.860	123.838
		123.850				
FORPNT	=$85	121.101*	121.463	121.466	121.468	121.471
		121.924	121.927	121.961	121.1036	121.1036
		121.1226	121.1341	121.1342	121.1357	121.1360
		121.1396	121.1400	121.1549	121.1550	121.1662
		121.1663	121.1682	121.1683	122.1378	122.1383
		122.1386	122.1387	123.435	123.436	
TXPSV	=$87	121.102*	121.1551	121.1551	121.1609	121.1609
LASTOP	=$87	121.103*	122.69	122.124		
CPRTYP	=$89	121.104*	122.33	122.43	122.44	122.46
		122.49	122.80	122.92	122.297	
FNCNAM	=$8A	121.105*	122.830	122.831	122.835	122.835
		122.838	122.838	122.840	122.844	122.855
		122.858	122.862	122.862	122.871	122.875
		122.989	122.1082	122.1082	122.1094	122.1101
		122.1108	122.1113			
TEMP3	=$8A	121.120*	123.923	123.929	123.939	123.1112
		123.1122				
DSCPTR	=$8C	121.106*	121.1386	121.1386	121.1390	121.1391
		121.1395	121.1399	122.889	122.889	122.1130
		122.1131	122.1219	122.1222	122.1229	122.1230
		122.1243	122.1260	122.1281	122.1281	
DSCLEN	=$8F	121.107*	122.1002	122.1014	122.1083	122.1084
		123.1231				
JMPADRS	=$90	121.108*	122.265	122.267	122.268	123.1213
LENGTH	=$91	121.109*	122.1083	122.1096	122.1100	122.1105
		122.1276	122.1282			
EXTRASV	=$92	121.117*	123.35	123.50	123.69	123.99
		123.969	123.990			
TEMP1	=$93	121.118*	121.535	121.538	121.542	123.432
		123.1006	123.1009	123.1115		
ARYPNT	=$94	121.110*	122.461	122.462	122.464	122.465
		122.497	122.498	122.644	122.646	122.649
		122.661	122.663	122.666	122.729	122.732
		122.1011	122.1012	122.1015	122.1016	122.1033
		122.1034	122.1037	122.1038	122.1050	122.1052
		122.1103	122.1103	122.1107	122.1110	122.1111
HIGHDS	=$94	121.111*	121.504	121.504	121.506	121.509
		121.513	121.515	121.642	121.647	124.823
		124.823				
HIGHTR	=$96	121.112*	121.489	121.491	121.497	121.500
		121.502	121.508	121.512	121.514	121.640
		121.644	122.456	122.456	122.1101	122.1102
TEMP2	=$98	121.119*	123.430	123.1035		
INDX	=$99	121.113*	122.749	122.770		
TMPEXP	=$99	121.121*	123.611	123.659	123.678	123.764
		123.776	123.779	123.784	123.796	123.844
EXPON	=$9A	121.122*	123.646	123.657	123.660	123.664
		123.668	123.698	123.708	123.715	123.795
		123.866	123.871			
LOWTR	=$9B	121.114*	121.489	121.492	121.600	121.603
		121.604	121.606	121.614	121.617	121.654
		121.815	121.816	121.817	121.822	121.829
		121.833	121.836	121.910	121.915	121.918

		121.937	121.940	121.943	121.944	121.945
		121.1220	121.1221	122.409	122.410	122.416
		122.420	122.424	122.455	122.455	122.471
		122.474	122.478	122.480	122.483	122.493
		122.494	122.553	122.554	122.560	122.565
		122.568	122.570	122.573	122.574	122.592
		122.608	122.613	122.622	122.634	122.637
		122.669	122.671	122.674	122.675	122.679
		122.690	122.695	122.743	122.746	122.991
		122.991	122.1075	122.1078	122.1080	122.1081
		122.1101	122.1102	124.188	124.188	124.196
		124.196	124.200	124.202	124.202	124.204
		124.204	124.227	124.230	124.243	124.246
		124.819	124.819	124.821		
DPFLG	=$9B	121.123*	123.648	123.649	123.676	
EXPSGN	=$9C	121.124*	123.639	123.642	123.702	
DSCTMP	=$9D	121.115*	124.703	124.708	124.731	124.734
FAC	=$9D	121.125*	121.129	121.535	121.542	121.544
		121.737	121.739	121.747	121.753	121.792
		121.795	121.798	121.953	121.954	121.963
		121.965	121.1019	121.1020	121.1282	121.1298
		121.1388	122.112	122.112	122.113	122.113
		122.114	122.137	122.171	122.273	122.277
		122.299	122.300	122.301	122.309	122.313
		122.326	122.507	122.538	122.540	122.687
		122.689	122.708	122.712	122.786	122.787
		122.889	122.889	122.891	122.892	122.893
		122.901	122.902	122.914	122.937	122.939
		122.941	122.944	122.945	122.1116	122.1116
		122.1123	122.1166	122.1167	122.1213	122.1248
		122.1263	122.1265	122.1310	122.1312	122.1347
		122.1351	122.1352	123.26	123.41	123.44
		123.51	123.63	123.71	123.72	123.73
		123.74	123.80	123.82	123.83	123.84
		123.85	123.86	123.87	123.89	123.95
		123.101	123.101	123.102	123.102	123.103
		123.103	123.104	123.104	123.109	123.110
		123.111	123.112	123.115	123.119	123.121
		123.123	123.124	123.125	123.126	123.131
		123.131	123.132	123.132	123.133	123.133
		123.134	123.134	123.138	123.140	123.142
		123.144	123.197	123.201	123.233	123.235
		123.237	123.239	123.289	123.295	123.302
		123.328	123.349	123.350	123.352	123.357
		123.360	123.363	123.366	123.385	123.386
		123.387	123.388	123.401	123.401	123.402
		123.402	123.412	123.415	123.418	123.423
		123.426	123.441	123.444	123.447	123.452
		123.455	123.464	123.472	123.482	123.490
		123.500	123.502	123.504	123.508	123.509
		123.510	123.530	123.534	123.538	123.542
		123.548	123.557	123.568	123.578	123.579
		123.580	123.585	123.595	123.596	123.600
		123.601	123.602	123.603	123.695	123.727
		123.728	123.753	123.812	123.815	123.816
		123.818	123.819	123.821	123.822	123.824
		123.946	123.971	123.985	123.986	123.1058

		123.1059	123.1060	123.1061	123.1064	123.1067
		123.1146				
VPNT	=$A0	121.116*	121.1356	121.1359	121.1365	121.1370
		121.1373	121.1377	121.1380	121.1381	121.1384
		122.207	122.208	122.217	122.220	122.251
		122.251				
pop	=$A1	121.26*	121.1240			
FACSGN	=$A2	121.129*	121.1017	121.1681	122.97	122.135
		122.315	122.321	122.505	123.23	123.23
		123.46	123.96	123.130	123.130	123.281
		123.306	123.309	123.421	123.450	123.461
		123.492	123.512	123.515	123.528	123.550
		123.561	123.576	123.590	123.591	123.693
		123.745	123.749	123.948	123.948	123.1063
		123.1090	123.1094	123.1119	123.1142	
SERLEN	=$A3	121.126*	123.617	123.670	123.1017	123.1037
FPGEN	=$A4	121.127*	123.159	123.565	123.572	123.582
		123.1227				
ARG	=$A5	121.128*	121.130	122.132	122.132	122.133
		122.133	122.134	122.134	122.272	122.275
		122.288	122.289	122.290	122.302	122.303
		122.305	122.306	122.325	123.36	123.37
		123.64	123.66	123.101	123.102	123.103
		123.104	123.251	123.252	123.253	123.254
		123.271	123.274	123.277	123.285	123.288
		123.292	123.356	123.359	123.362	123.365
		123.377	123.378	123.379	123.380	123.385
		123.385	123.386	123.386	123.387	123.387
		123.388	123.388	123.463	123.473	123.919
		123.984	123.987			
ARGSGN	=$AA	121.130*	122.286	123.24	123.45	123.280
		123.283	123.460	123.692	123.926	123.1080
goto	=$AB	121.27*	121.1278	121.1296	124.151	
SGNCPR	=$AB	121.131*	122.136	123.25	123.61	123.282
		123.305	123.326	123.340	123.694	123.998
		123.1085				
STRNG1	=$AB	121.134*	121.1386	121.1386	122.212	122.899
		122.900	122.905	122.916	122.918	122.922
		122.928	122.929	122.1119	122.1119	122.1121
		122.1134	122.1135	122.1141	122.1144	122.1147
EXTRAFAC	=$AC	121.132*	123.34	123.54	123.58	123.70
		123.88	123.90	123.100	123.108	123.127
		123.135	123.135	123.136	123.152	123.166
		123.231	123.259	123.396	123.427	123.457
		123.467	123.476	123.484	123.511	123.546
		123.589	123.965	123.991	123.1065	
SERPNT	=$AD	121.133*	123.1003	123.1004	123.1013	123.1014
		123.1016	123.1018	123.1022	123.1023	123.1024
		123.1026	123.1027	123.1032	123.1033	
STRNG2	=$AD	121.135*	121.735	121.766	122.605	122.617
		122.639	122.640	122.657	122.658	122.664
		122.682	122.683	122.702	122.703	122.713
		122.716	122.759	122.760	122.917	122.921
		122.1318	122.1318	122.1339	122.1339	123.750
		123.800	123.809	123.839	123.849	123.857
PRGEND	=$AF	121.136*	121.593	121.593	121.662	121.662
		121.851	121.855	121.1149	121.1149	124.185

		124.185				
gosub	=$B0	121.28*	121.1197	121.1229	121.1294	
CHRGET	=$B1	121.137*	121.585	121.898	121.1032	121.1075
		121.1091	121.1302	121.1337	121.1453	121.1569
		121.1708	122.47	122.145	122.196	122.226
		122.242	122.369	122.374	122.394	122.503
		122.1307	123.621	123.627	123.640	123.1221
rem	=$B2	121.29*	121.779			
CHRGOT	=$B7	121.138*	121.892	121.1029	121.1199	121.1277
		121.1287	121.1404	121.1553	121.1603	121.1610
		121.1705	122.34	122.346	122.351	122.354
		122.543	122.863	122.1249	122.1313	122.1334
		122.1380	124.172	124.726	124.766	
TXTPTR	=$B8	121.139*	121.582	121.583	121.706	121.743
		121.790	121.803	121.805	121.880	121.881
		121.1007	121.1009	121.1042	121.1043	121.1050
		121.1053	121.1057	121.1060	121.1063	121.1063
		121.1063	121.1122	121.1123	121.1144	121.1145
		121.1195	121.1195	121.1210	121.1211	121.1220
		121.1221	121.1245	121.1245	121.1249	121.1249
		121.1249	121.1262	121.1509	121.1509	121.1551
		121.1551	121.1552	121.1552	121.1567	121.1568
		121.1575	121.1587	121.1588	121.1608	121.1608
		121.1609	121.1609	121.1624	121.1627	121.1630
		121.1633	121.1697	121.1699	122.23	122.23
		122.23	122.78	122.78	122.78	122.158
		122.159	122.194	122.816	122.816	122.854
		122.854	122.856	122.859	122.867	122.867
		122.1318	122.1318	122.1320	122.1325	122.1339
		122.1339	123.712	123.1184	123.1184	124.153
		124.153	124.170	124.170	124.177	124.177
print	=$BA	121.30*	121.729			
tab	=$C0	121.31*	121.1407			
to	=$C1	121.32*	121.1013	124.722	124.727	
fn	=$C2	121.33*	122.178	122.825		
spc	=$C3	121.34*	121.1409			
then	=$C4	121.35*	121.1280			
at	=$C5	121.36*	121.758	121.765	124.59	124.767
not	=$C6	121.37*	122.166			
step	=$C7	121.38*	121.1030			
plus	=$C8	121.39*	122.52	122.154	123.633	
minus	=$C9	121.40*	121.886	121.894	122.152	123.629
RNDSEED	=$C9	121.140*	123.1047	123.1069		
equal	=$D0	121.41*	121.1343	122.812		
DXL	=$D0	121.141*	124.474	124.489	124.519	124.594
		124.611	124.646	124.663		
DXH	=$D1	121.142*	124.370	124.382	124.392	124.471
		124.492	124.522	124.604	124.623	124.656
		124.673				
sgn	=$D2	121.42*	122.182			
DY	=$D2	121.143*	124.485	124.500	124.599	124.617
		124.651	124.668			
QDRNT	=$D3	121.144*	124.393	124.463	124.470	124.487
		124.514	124.589	124.641		
EL	=$D4	121.145*	124.475	124.499	124.501	124.518
		124.520	124.606	124.609	124.612	124.658
		124.661	124.664			

EH	=$D5	121.146*	124.472	124.502	124.504	124.521
		124.607	124.616	124.618	124.659	124.668
		124.668				
LOCK	=$D6	121.147*	121.710	121.843	121.1157	121.1160
		121.1172	123.1265			
scrn	=$D7	121.43*	122.237			
ERRFLG	=$D8	121.148*	121.558	121.584	121.1113	121.1502
		124.155				
ERRLIN	=$DA	121.149*	124.168	124.168	124.176	124.176
ERRPOS	=$DC	121.150*	124.169	124.169	124.177	124.177
ERRNUM	=$DE	121.151*	124.162			
ERRSTK	=$DF	121.152*	124.164	124.178		
XOL	=$E0	121.153*	124.285	124.453	124.459	124.477
		124.578				
XOH	=$E1	121.154*	124.286	124.454	124.462	124.478
		124.572	124.577			
YO	=$E2	121.155*	124.284	124.455	124.481	124.486
		124.554	124.559	124.560		
HCOLORZ	=$E4	121.156*	124.322	124.717		
HNDX	=$E5	121.157*	124.316	124.349	124.494	124.561
		124.563	124.575	124.600	124.652	
HPAG	=$E6	121.158*	124.262	124.267	124.306	124.443
SCALEZ	=$E7	121.159*	124.608	124.660	124.743	
leftstr	=$E8	121.44*	122.243			
SHAPEPNT	=$E8	121.160*	124.747	124.747	124.763	124.809
		124.811				
COLCOUNT	=$EA	121.161*	124.377	124.389	124.602	124.654
FIRST	=$F0	121.162*	122.229	124.40	124.53	124.55
		124.58	124.68	124.78	124.86	
SPEEDZ	=$F1	121.163*	121.1486	123.1222	124.107	
TRCFLG	=$F2	121.164*	121.1064	123.1225	124.113	
ORMASK	=$F3	121.165*	121.1482	124.121		
TXTPSV	=$F4	121.166*	124.153	124.153	124.170	124.170
CURLSV	=$F6	121.167*	124.156	124.156	124.171	124.171
REMSTK	=$F8	121.168*	121.1040	124.163		
ROTZ	=$F9	121.169*	124.739	124.772		
STACK	=$0100	121.174*	121.460	121.465	121.467	121.469
		121.472	121.1680	121.1690	121.1692	121.1694
		121.1696	121.1698	122.444	122.534	122.536
		122.539	122.541	123.748	123.803	123.808
		123.843	123.848	123.858	123.874	123.876
		123.884	123.886	123.888	123.890	123.892
		123.893	123.894			
IN	=$0200	121.175*	121.636	121.649	121.649	121.652
		121.689	121.692	121.694	121.698	121.699
		121.718	121.749	121.760	121.769	121.770
		121.782	121.787	121.800	121.802	121.1513
		121.1514	121.1516	121.1531	121.1533	121.1559
		121.1560	121.1561	121.1597	123.1234	123.1235
AMPER	=$03F5	121.176*	121.253			
KEY	=$C000	121.177*	121.1107			
TXTCLR	=$C050	121.178*	124.264			
MIXCLR	=$C052	121.179*	124.256			
MIXSET	=$C053	121.180*	124.216	124.261		
LOWSCR	=$C054	121.181*	124.219	124.260		
HISCR	=$C055	121.182*	124.255			
LORES	=$C056	121.183*	124.215			

	HIRES	=$C057	121.184*	124.263		
MD	INCR	=$D000	123.1184	123.1268	124.194	124.202	124.203
			124.628	124.678			
	DECR	=$D000	122.23	122.78			
MD	CMPR	=$D000	124.130	124.144	124.146	124.196	124.204
MD	TR	=$D000	121.602	121.603	121.662	121.662	121.1157
			122.1083	123.401	123.401	123.402	123.402
			124.148	124.148	124.153	124.153	124.156
			124.156	124.168	124.168	124.169	124.169
			124.170	124.170	124.171	124.171	124.176
			124.176	124.177	124.177	124.188	124.188
			124.747	124.747	124.823	124.823	
MD	TRDB	=$D000	121.662	123.401	123.402	124.148	124.153
			124.156	124.168	124.169	124.170	124.171
			124.176	124.177	124.188	124.747	124.823
MD	TRX	=$D000	121.593	124.185			
MD	TRAX	=$D000	122.991	122.1003	122.1082	122.1103	
MD	TRAY	=$D000	121.638	121.649	121.650	121.657	121.859
			121.860	121.1129	121.1146	121.1175	121.1176
			121.1386	121.1509	121.1551	121.1608	121.1609
			122.455	122.456			
MD	TRXY	=$D000	121.1552	122.889	122.1318	122.1339	
MD	AD	=$D000	121.880	121.881	121.1156	121.1156	121.1331
			121.1334	122.1101	122.1102	123.101	123.102
			123.103	123.104	123.251	123.252	123.253
			123.254	124.668	124.819	124.819	
MD	ADD	=$D000	121.1156	124.819			
MD	SB	=$D000	121.489	121.504	121.1149	121.1149	121.1220
			121.1221	123.71	123.72	123.73	123.74
			123.385	123.386	123.387	123.388	
MD	SUB	=$D000	121.1149				
MD	BUMP	=$D000	121.1063	121.1249	122.1048	122.1086	122.1162
			122.1190	122.1236			
MD	PUSH	=$D000	121.1012	121.1036	121.1195	121.1196	121.1345
			122.112	122.113	122.251	122.528	122.815
			122.816	122.835	122.854	122.860	122.1116
MD	PULL	=$D000	121.1245	122.132	122.133	122.134	122.530
			122.548	122.838	122.862	122.867	122.1119
			122.1281				
MD	NEG	=$D000	123.23	123.130	123.131	123.132	123.133
			123.134	123.135	123.948	123.1096	
	CMDTABL	=$D000	121.206*	121.1087	121.1089		
	UNFNC	=$D080	121.270*	122.264	122.266		
	MATHTBL	=$D0B2	121.298*	122.63	122.79	122.83	122.87
			122.89	122.98	122.168	122.201	
	MINUS	=$D0C7	121.312*	122.201			
	EQUAL	=$D0CA	121.314*	122.168			
	PLUS	=$D0CD	121.316*	122.79			
	TOKTABL	=$D0D0	121.319*	121.736	121.738	121.962	121.964
	ERRMSG	=$D260	121.428*	121.557	121.564	121.1139	121.1231
			121.1233	121.1622	121.1666	121.1720	122.198
			122.576	122.578	122.581	122.598	122.799
			122.801	122.934	122.976	122.1125	123.147
			123.399				
	NXwoFOR	=$D260	121.429*	121.1666			
	SYNTXERR	=$D270	121.430*	122.198			
	RTNwoGSB	=$D276	121.431*	121.1231			

OofDATA	=$D28A	121.432*	121.1622	122.598	
ILLQUAN	=$D295	121.433*	122.578		
OVFLOW	=$D2A5	121.434*	123.147		
OofMEM	=$D2AD	121.435*	121.557	122.976	
UNDSTAT	=$D2BA	121.436*	121.1233		
BADSUBS	=$D2CB	121.437*	122.576		
REdimARR	=$D2D8	121.438*	122.581		
DIVbyZRO	=$D2E5	121.439*	123.399		
ILLDIR	=$D2F5	121.440*	122.799		
TYPEMISS	=$D303	121.441*	121.1720		
STRtoLNG	=$D310	121.442*	122.1125		
FORMtoCX	=$D31F	121.443*	122.934		
CANTCON	=$D332	121.444*	121.1139		
UNDFUNC	=$D340	121.445*	122.801		
ERRIN	=$D350	121.447*	121.571	121.572	
INMSG	=$D358	121.449*	123.722	123.723	
BREAKIN	=$D35D	121.452*	121.1132	121.1133	
GTFORPNT	=$D365	121.456*	121.994	121.1227	121.1664
FNDFOR	=$D36A	121.460*	121.478		
SAMEFOR?	=$D37F	121.464	121.469*		
NXFOR	=$D38B	121.470	121.474*		
RET1	=$D392	121.462	121.473	121.479*	
BLTU	=$D393	121.481*	121.648	122.463	
BLTU2	=$D39A	121.488*	122.1104		
SETEND	=$D3B7	121.501	121.504*		
MVBYT	=$D3C3	121.508*	121.511		
NXBYT	=$D3C7	121.505	121.507	121.510*	121.517
NXPAG	=$D3CE	121.496	121.514*		
CHKMEM	=$D3D6	121.520*	121.1003	121.1194	122.30
REASON	=$D3E3	121.481	121.529*	122.602	122.653 123.1271
RS1	=$D3ED	121.531	121.534*		
RS2	=$D3F1	121.537*	121.540		
RS3	=$D3FC	121.543*	121.546		
RET2	=$D40F	121.530	121.533	121.551	121.555*
MEMERR	=$D410	121.522	121.526	121.552	121.554 121.557*
		122.699	124.132	124.249	124.805
ERROR	=$D412	121.275	121.558*	121.1142	121.1234 121.1721
		122.199	122.579	122.599	122.802 122.935
		122.1126	123.148	123.400	
DOERRMSG	=$D419	121.559	121.562*		
ERLUP	=$D41F	121.564*	121.569		
PRNTIN?	=$D431	121.573*	121.1135		
RESTART	=$D43C	121.576	121.579*	121.587	121.663 121.1136
		122.366	123.1281	123.1282	
NXLIN	=$D45C	121.590	121.593*		
NL1	=$D49F	121.620	121.623*		
MVDWN	=$D4A7	121.625	121.628*	121.631	121.635
NEWLN?	=$D4B5	121.598	121.636*		
MVPRG	=$D4D1	121.645	121.647*		
INSRTLIN	=$D4EA	121.652*	121.655		
LINKSET	=$D4F2	121.637	121.656*	121.1163	124.213
NXLINK	=$D4FE	121.659*	121.680		
PUTLINK	=$D50F	121.661	121.664*		
FINDEOL	=$D511	121.665*	121.667		
INLIN	=$D52C	121.682*	121.1532	121.1539	
INLIN2	=$D52E	121.581	121.683*		
GDBUFS	=$D539	121.686	121.688*		

```
STRIP    =$D541  121.692*    121.696
NOI      =$D54C  121.691     121.697*
INCHR    =$D553  121.702*    121.1111
GETIN    =$D559  121.591     121.595     121.706*
PARSE    =$D56C  121.711     121.717*    121.722
NXCHR    =$D56D  121.718*    121.780
SE       =$D578  121.720     121.723*
TOK?     =$D588  121.728     121.731*
ISTOK?   =$D590  121.732     121.735*
NY       =$D5A2  121.745*    121.754
NX       =$D5A7  121.746     121.748*    121.751
LIN      =$D5A8  121.749*    121.799
PUTTOK   =$D5CB  121.759     121.766*    121.801
PUTIN    =$D5CD  121.726     121.730     121.734     121.767*    121.783
                 121.785
SSF      =$D5E0  121.774     121.777*
REM?     =$D5E2  121.776     121.778*
SHFTIN   =$D5E9  121.782*    121.789
SHIN     =$D5F2  121.725     121.786*
SKIPTOK  =$D5F9  121.756     121.762     121.764     121.790*
SK2      =$D5FD  121.792*    121.797
PLU?     =$D604  121.794     121.796*
DONE     =$D610  121.771     121.802*
FNDLIN   =$D61A  121.597     121.812*    121.891     124.187     124.195
FL1      =$D61E  121.814*    121.837     121.1217
FL2      =$D635  121.824     121.827*
GETLINK  =$D63E  121.826     121.832*
NOSUCH   =$D647  121.818     121.838*
RET3     =$D648  121.823     121.830     121.831     121.839*    121.841
                 121.897
NEW      =$D649  121.269     121.841*
SCRTCH   =$D64B  121.842*    123.1272
SETPTRS  =$D665  121.656     121.714     121.856*    121.1162    121.1183
CLEAR    =$D66A  121.267     121.858*
CLEARC   =$D66C  121.859*    121.1184    124.149
STKINI   =$D683  121.570     121.864*
RET4     =$D696  121.858     121.877*    121.889     121.900
STXTPT   =$D697  121.856     121.879*
LIST     =$D6A5  121.266     121.884*
STRTRNG  =$D6B1  121.884     121.885     121.887     121.890*
ENDRNG   =$D6C4  121.895     121.898*
MAINLST  =$D6CC  121.893     121.901*
NXLST    =$D6DA  121.905     121.909*    121.946
LSTD?    =$D6F5  121.920     121.923*
LST1LIN  =$D6F7  121.922     121.924*
LISTLOOP=$D6FE  121.927*    121.980
SENDCHR  =$D702  121.929*    121.957
NCR      =$D712  121.932     121.936*
LISTED   =$D724  121.911     121.923     121.947*
GETCHR   =$D72C  121.951*    121.969     121.974
GC       =$D731  121.952     121.954*
TOKEN?   =$D734  121.938     121.957*
SKPTK    =$D746  121.967*    121.971
TOKLP    =$D749  121.969*    121.970
PRTOK    =$D750  121.968     121.972*
TOKLUP   =$D755  121.974*    121.977
TOKDONE  =$D75F  121.975     121.978*
```

```
FOR       =$D766  121.207     121.991*
FOR2      =$D777  121.995     121.1000*
STEP      =$D7AF  121.1021    121.1022    121.1026*
ONESTEP   =$D7C3  121.1031    121.1034*
NEWSTT    =$D7D2  121.715     121.949     121.1039*   121.1077    121.1201
                  121.1700    124.174     124.180
DIRCT     =$D7E5  121.1046    121.1049*
TRACE?    =$D805  121.592     121.1064*   121.1094
TR1       =$D81D  121.1065    121.1068    121.1075*
GOEND     =$D826  121.1055    121.1079*
GOCMD     =$D828  121.1076    121.1080*   121.1290
GOCMD2    =$D82A  121.1081*   121.1301
NOTOK     =$D83F  121.1082    121.1092*
COLON?    =$D842  121.1051    121.1093*
JSY       =$D846  121.1084    121.1095*
RESTORE   =$D849  121.252     121.863     121.1097*
SETDA     =$D853  121.1101    121.1103*   121.1643
RET5      =$D857  121.1080    121.1105*
ISCNTC    =$D858  121.912     121.1041    121.1107*
GK        =$D860  121.1109    121.1111*
ERFLG?    =$D863  121.1112*   121.1536
CTRC?     =$D86C  121.1114    121.1116*
STOP      =$D86E  121.257     121.1118*
END       =$D870  121.206     121.1120*
END2      =$D871  121.1118    121.1121*
END3      =$D888  121.1126    121.1130*
END4      =$D88A  121.1079    121.1132*
GOSTART   =$D893  121.1134    121.1136*
CONT      =$D896  121.265     121.1138*
CON       =$D8A1  121.1141    121.1143*
RET6      =$D8AF  121.1121    121.1138    121.1147*
SAVE      =$D8B0  121.261     121.1149*
LOAD      =$D8C9  121.260     121.1154*
JLNK      =$D8ED  121.1161    121.1163*
VARTIO    =$D8F0  121.1150    121.1154    121.1165*   124.833
PROGIO    =$D901  121.1152    121.1158    121.1175*
RUN       =$D912  121.250     121.1179*
RUNLINE   =$D91B  121.1182    121.1184*
GOSUB     =$D921  121.254     121.1193*
GOLINE    =$D935  121.1185    121.1199*
GOTO      =$D93E  121.249     121.1200    121.1203*   121.1289    124.173
GO1       =$D955  121.1207    121.1215*
GO2       =$D959  121.1212    121.1214    121.1217*
RET7      =$D96A  121.1222*   121.1224
POP       =$D96B  121.239     121.255     121.1224*
UNDERR    =$D97C  121.1218    121.1233*
GSYNER    =$D981  121.1236*   121.1297
RETURN    =$D984  121.1230    121.1238*
DATA      =$D995  121.209     121.1246*   122.817
ADDON     =$D998  121.1247*   121.1285    121.1635    124.158
RET8      =$D9A2  121.1250*   121.1263    121.1265
DATAN     =$D9A3  121.1004    121.1246    121.1252*   121.1618
REMN      =$D9A6  121.1204    121.1254*   121.1284    124.157
RM1       =$D9AE  121.1258*   121.1269
RM2       =$D9B6  121.1262*   121.1268
PULL3     =$D9C5  121.1241    121.1271*
IF        =$D9C9  121.251     121.1276*
```

TRUE? =$D9D8 121.1279 121.1282*
REM =$D9DC 121.256 121.1284*
IFTRUE =$D9E1 121.1283 121.1287*
JGOCMD =$D9E9 121.1288 121.1290*
ONGOTO =$D9EC 121.258 121.1292*
GOTO? =$D9F4 121.1296* 121.1318
ONCNT =$D9F8 121.1295 121.1298* 121.1305
NXNUM =$DA00 121.1299 121.1302*
RET9 =$DA0B 121.1307* 121.1312
LINGET =$DA0C 121.594 121.890 121.899 121.1203 121.1303
121.1309* 124.186 124.191
ASCHEX =$DA12 121.1312* 121.1338
NXDIG =$DA40 121.1335 121.1337*
LET =$DA46 121.248 121.993 121.1092 121.1340*
LET2 =$DA63 121.1352* 121.1602
LETREAL =$DA77 121.1352 121.1362*
LETSTR =$DA7A 121.1350 121.1363*
PUTSTR =$DA7B 121.1364* 121.1594
DESC? =$DA8C 121.1368 121.1373*
COPSTR =$DA9A 121.1367 121.1372 121.1375 121.1380*
NEWDESC =$DAA1 121.1376 121.1379 121.1383*
COPY =$DAB7 121.1382 121.1390*
PRSTRING=$DACF 121.1403* 121.1419 121.1422
PRINT =$DAD5 121.264 121.1405*
PRINT2 =$DAD7 121.1406* 121.1454
CRDO =$DAFB 121.562 121.579 121.913 121.933 121.1405
121.1424* 121.1432 123.1232 124.845
NEGATE =$DB00 121.1426* 121.1469
RET10 =$DB02 121.1406 121.1427* 121.1463
TAB =$DB03 121.1414 121.1429*
NXCLM =$DB0E 121.1431 121.1434*
TABWHERE=$DB16 121.1408 121.1411 121.1438*
SPC? =$DB21 121.1441 121.1443*
TABIT =$DB2B 121.1444 121.1450*
NXSPC =$DB2C 121.1451* 121.1456
NEXTCHR =$DB2F 121.1416 121.1433 121.1437 121.1448 121.1453*
DOSPC =$DB35 121.1452 121.1455*
STROUT =$DB3A 121.573 121.1457* 121.1508 121.1649 123.733
123.1277 123.1278
STRPRT =$DB3D 121.1403 121.1458* 121.1526
NXCHAR =$DB44 121.1462* 121.1468 121.1470
OUTSP =$DB57 121.1074 121.1455 121.1476*
OUTQUES =$DB5A 121.563 121.1478* 121.1528 121.1538 121.1565
OUTDO =$DB5C 121.566 121.929 121.948 121.973 121.976
121.978 121.1070 121.1425 121.1465 121.1479*
SEND =$DB64 121.1481 121.1483*
INPUTERR=$DB71 121.1491* 121.1607
READERR =$DB7B 121.1493 121.1496*
ERLIN =$DB7F 121.1495 121.1498*
INPERR =$DB86 121.1501* 121.1616
RESPERR =$DB87 121.1492 121.1502*
DOREENT =$DB90 121.1503 121.1506*
GET =$DBA0 121.268 121.1512*
INPUT =$DBB2 121.210 121.1521*
QOUT =$DBC4 121.1522 121.1528*
DIR? =$DBC7 121.1527 121.1529*
NXIN =$DBDC 121.1538* 121.1566

READ =$DBE2 121.213 121.1540*
ZF =$DBE9 121.1535 121.1544*
MAININP =$DBEB 121.1518 121.1545*
NXINP =$DBF1 121.1548* 121.1613
SNDQ? =$DC1F 121.1556 121.1564*
STXP =$DC27 121.1562 121.1567*
INSTART =$DC2B 121.1554 121.1569* 121.1638
PUTCHR =$DC3F 121.1573 121.1579*
PENCHR =$DC4B 121.1578 121.1585*
PECHR =$DC4C 121.1581 121.1586*
SKP =$DC57 121.1590 121.1592*
NUMIN =$DC63 121.1571 121.1596*
DATIN =$DC69 121.1599* 121.1615
WNX =$DC72 121.1595 121.1603*
SWPNT =$DC7E 121.1604 121.1606 121.1608*
INPFIN =$DC99 121.1598 121.1614*
FINDATA =$DCA0 121.1564 121.1618* 121.1637
NXS =$DCB9 121.1621 121.1633*
INPDONE =$DCC6 121.1611 121.1639*
NTD =$DCD1 121.1642 121.1644*
RET11 =$DCDE 121.1646 121.1650*
EXIG =$DCDF 121.1647 121.1648 121.1652*
REENT =$DCEF 121.1506 121.1507 121.1655*
NEXT =$DCF9 121.208 121.1658*
VARNXT =$DCFF 121.1658 121.1661* 121.1709
SKPV =$DD02 121.1660 121.1662*
GERR =$DD0D 121.1625 121.1667*
GOTFOR =$DD0F 121.1665 121.1668*
GONEWST =$DD52 121.1700* 121.1707
ENDFOR =$DD55 121.1691 121.1701*
FRMNUM =$DD67 121.1016 121.1033 121.1711* 122.504 122.861
122.1308 122.1342 124.18 124.128 124.142
124.685
CHKNUM =$DD6A 121.1015 121.1712* 122.65 122.123 122.270
122.810 122.832 122.837 122.878
CHKSTR =$DD6C 121.1714* 122.248 122.1118 122.1165
CHKVAL =$DD6D 121.1349 121.1715* 122.284
RET12 =$DD73 121.1718* 121.1719
CV2 =$DD74 121.1716 121.1719*
MISMTCH =$DD76 121.1717 121.1720*
JERROR =$DD78 121.1667 121.1721*
FRMEVL =$DD7B 121.1276 121.1346 121.1417 121.1711 122.23*
122.187 122.246
FEVLOOP =$DD86 122.26* 122.93
FRMEVL2 =$DD95 122.34* 122.1138
CPROP =$DD98 122.35* 122.48
CHKTYP =$DDB4 122.37 122.39 122.49*
ARITH =$DDC5 122.55 122.57*
PREFTEST=$DDCD 122.62* 122.81
NXOP =$DDD6 122.66* 122.85
SAVOP =$DDD7 122.67* 122.204
COMPARE =$DDE4 122.50 122.75*
MND =$DDEE
PREFNC =$DDF6 122.70 122.83*
PSHMAD =$DDFD 122.67 122.87*
SNTXERR =$DE0D 122.45 122.95*
PSHF =$DE10 122.91 122.97*

PSHFACX =$DE15 121.1035 122.103*
PUSHFAC =$DE20 121.1025 122.111*
NOTMATH =$DE35 122.51 122.53 122.118*
GOEX =$DE38 122.72 122.120*
DOMTH =$DE3A 122.64 122.121*
DMTH =$DE41 122.122 122.124*
DOMATH =$DE43 122.73 122.84 122.129*
EXIT =$DE5D 122.120 122.137*
GETVAL =$DE60 122.31 122.143* 122.1117
SKIP =$DE64 122.145* 122.155
NUMBER =$DE69 122.147* 122.151
VAR? =$DE6C 122.146 122.148*
STRTXT =$DE81 121.1523 122.158*
ST1 =$DE8A 122.161 122.163*
NOT? =$DE90 122.157 122.166*
EQUOP =$DE98 121.315 122.171*
NOTZ =$DE9F 122.172 122.175*
FN? =$DEA4 122.167 122.178*
SGN? =$DEAB 122.179 122.182*
PARCHK =$DEB2 122.183 122.186* 122.261 122.836
CHKCLS =$DEB8 122.188* 122.233 122.547 122.811 122.1272
CHKOPN =$DEBB 122.186 122.190* 122.245 122.806
CHKCOM =$DEBE 121.1612 122.192* 122.247 122.343 122.1252
122.1344
SYNCHR =$DEC0 121.1014 121.1281 121.1344 121.1525 122.193*
122.813 122.826 124.42 124.60 124.152
124.190 124.699 124.729 124.769
SYNERR =$DEC9 121.1095 121.1236 121.1442 121.1500 122.95
122.195 122.198* 122.357 122.865 124.182
MIN =$DECE 122.153 122.201*
EQUL =$DED0 122.169 122.202*
VARL =$DED5 122.149 122.206* 122.441 122.445
VR1 =$DEE5 122.210 122.214*
VR2 =$DEF6 122.215 122.224*
SCREEN =$DEF9 122.226* 122.238
UNARY =$DF0C 122.184 122.237*
NOTinstr=$DF3A 122.244 122.261*
GOROUT =$DF3F 122.259 122.264*
OR =$DF4F 121.311 122.272*
AND =$DF55 121.309 122.275*
FALSE =$DF5D 122.276 122.279*
TRUE =$DF60 122.274 122.278 122.281*
POSOP =$DF65 121.317 122.284*
STRCMP =$DF7D 122.285 122.295*
SFS =$DFA5 122.310 122.312 122.315*
CMPLOOP =$DFAA 122.318* 122.327
NUMCMP =$DFB0 122.294 122.322*
DOCMP =$DFB5 122.320 122.325*
CMPDONE =$DFC1 122.322 122.324 122.329 122.331*
JF =$DFCA 122.335 122.337*
PDL =$DFCD 121.276 122.339*
NXDIM =$DFD6 122.343* 122.347
DIM =$DFD9 121.212 122.344*
PTRGET =$DFE3 121.1340 121.1548 121.1661 122.206 122.350*
122.809 124.830
PTRGET2 =$DFE8 122.345 122.352*
PTRGET3 =$DFEA 122.353* 122.829

BADNAM =$DFF4 122.357* 122.386
NAMOK =$DFF7 122.356 122.358*
MORNAM =$E007 122.361 122.369*
GTLT =$E011 122.370 122.373*
BYPASS =$E012 122.374* 122.375 122.377
STRNG? =$E01C 122.372 122.378*
INTVAR? =$E026 122.379 122.383*
NIN =$E036 122.382 122.391*
SCDCH =$E03D 122.384 122.395*
JARY =$E046 122.400* 122.403
BSB =$E049 122.399 122.401*
VSEARCH =$E04F 122.402 122.404*
NXVAR =$E059 122.409* 122.428
NV1 =$E05B 122.410* 122.426
NV2 =$E065 122.412 122.415*
NXPTR =$E073 122.417 122.423*
ISLETC =$E07D 122.148 122.355 122.371 122.376 122.432*
RTN1 =$E086 122.433 122.437*
NOTFND =$E087 122.414 122.439*
TWOBRK =$E09A 122.447 122.448 122.450*
NEWVAR =$E09C 122.442 122.446 122.455*
NWV =$E0B2 122.459 122.461*
SETVPNT =$E0DE 122.421 122.480*
SVP =$E0E8 122.484 122.486*
GETARY =$E0ED 122.490* 122.589 122.601
GETARY2 =$E0EF 122.491* 124.822
GD =$E0F9 122.495 122.497*
NEGNUM =$E0FE 122.501* 122.510 122.511
MAKINT =$E102 122.503* 122.529
MKINT =$E108 122.505* 122.1309
AYINT =$E10C 121.1354 122.507*
MI1 =$E119 122.506 122.513*
MI2 =$E11B 122.509 122.514*
ARRAY =$E11E 122.400 122.518*
NXTDIM =$E12C 122.526* 122.545
FNDARY =$E169 122.519 122.551*
ARYLOOP =$E16D 122.553* 122.575
ARYNAM? =$E179 122.556 122.559*
NXARY =$E188 122.563 122.567*
SUBERR =$E196 122.576* 122.593 122.697
IQERR =$E199 122.513 122.578* 122.1305 123.196 123.1215
123.1216 124.50
JER =$E19B 122.579* 122.583
ARYFOUND=$E19E 122.566 122.581*
CHKDIM =$E1AA 122.585 122.589*
NOTFOUND=$E1B8 122.558 122.596*
MAKARY =$E1C1 122.597 122.601*
NINT =$E1D5 122.609 122.611*
RAR =$E1DE 122.614 122.617*
SAVDIM =$E1E7 122.623* 122.643
DFLTDIM =$E1F7 122.626 122.633*
ZARY =$E21A 122.650 122.653*
ZLUP =$E229 122.660* 122.662 122.665
NXPG =$E22E 122.659 122.663*
FNDELEM =$E24B 122.594 122.679*
DIMLUP =$E253 122.683* 122.715
GSE =$E269 122.692 122.697*

GME	=$E26C	122.645	122.652	122.699*	122.758	122.769
DIMOK	=$E26F	122.691	122.701*			
DIMOK2	=$E270	122.696	122.702*			
NXDM	=$E281	122.705	122.712*			
NINTA	=$E292	122.719	122.721*			
RARY	=$E298	122.722	122.725*			
RTN2	=$E2AC	122.677	122.736*	122.798		
MULT	=$E2AD	122.638	122.706	122.742*		
MU1	=$E2B6	122.727	122.747*			
MU2	=$E2C0	122.752*	122.771			
MU3	=$E2D9	122.761	122.770*			
FRE	=$E2DE	121.274	122.774*			
FRE2	=$E2E5	122.775	122.777*			
GIVAYF	=$E2F2	122.223	122.784*	122.794		
POS	=$E2FF	121.277	122.791*			
SNGFLT	=$E301	122.176	122.232	122.282	122.341	122.792*
		122.1292	122.1304	122.1369		
ERRDIR	=$E306	121.1512	121.1529	122.796*	122.805	
UNDFNC	=$E30E	122.801*	122.845			
DEF	=$E313	121.262	122.804*			
FNC?	=$E341	122.804	122.825*	122.834		
FUNCT	=$E354	122.180	122.834*			
SAVOLD	=$E378	122.848*	122.851			
GETOLD	=$E3A9	122.864	122.867*			
FNCDATA	=$E3AF	122.818	122.868*			
STR	=$E3C5	121.288	122.878*			
STRINI	=$E3D5	121.1385	122.889*	122.927	122.1128	
STRSPA	=$E3DD	122.890*	122.1210	122.1228		
STRLIT	=$E3E7	121.1421	121.1457	122.163	122.885	122.896*
STRLT2	=$E3ED	121.1592	122.899*			
FEND	=$E3F7	122.904*	122.910			
QUO?	=$E404	122.908	122.911*			
ZEND	=$E408	122.906	122.913*			
NZ	=$E409	122.912	122.914*			
FE1	=$E415	122.919	122.921*			
FE2	=$E41F	122.923	122.926*			
PUTNEW	=$E42A	122.925	122.931*	122.1137	122.1216	122.1239
JERR	=$E432	122.935*	122.978			
PUTEMP	=$E435	122.933	122.937*			
GETSPA	=$E452	122.890	122.955*			
GETSPC	=$E454	122.956*	122.983			
CY	=$E45F	122.961	122.963*			
GOTSPA	=$E469	122.965	122.968*			
FULL	=$E474	122.964	122.967	122.976*		
GARBAG	=$E484	121.541	122.777	122.979	122.984*	
FNDVAR	=$E488	122.986*	122.1114			
TVAR	=$E4A0	122.996*	122.999			
SVARS	=$E4A9	122.997	122.1001*			
SVAR	=$E4B5	122.1004*	122.1009			
SVARGO	=$E4BD	122.1005	122.1008*			
ARYVAR	=$E4C2	122.1007	122.1011*			
ARYVA2	=$E4CA	122.1015*	122.1040	122.1042		
ARYVA3	=$E4CE	122.1017*	122.1053			
ARYVGO	=$E4D9	122.1018	122.1020	122.1023*		
ARYSTR	=$E50C	122.1050*	122.1055			
GOGO	=$E514	122.1051	122.1054*			
DVARS	=$E519	122.1008	122.1057*			

```
   DVAR      =$E523  122.998     122.1054    122.1063*
   DV1       =$E538  122.1071    122.1075*
   DV2       =$E542  122.1077    122.1080*
   DVARTS    =$E552  122.1058    122.1061    122.1064    122.1072    122.1074
                     122.1076    122.1079    122.1084*
   VDONE     =$E55D  122.1087*   122.1095
   GRBPAS    =$E562  122.1021    122.1094*
   CAT       =$E597  122.56      122.1116*
   NTL       =$E5B7  122.1124    122.1128*
   MOVINS    =$E5D4  121.1387    122.1129    122.1140*
   MOVSTR    =$E5E2  122.930     122.1150*
   MOVESTR   =$E5E6  122.1133    122.1152*   122.1238
   MVS2      =$E5EA  122.1155*   122.1159
   MVS3      =$E5F3  122.1153    122.1161*
   FRESTR    =$E5FD  122.1165*   122.1293
   FREFAC    =$E600  121.1458    122.298     122.776     122.1166*
   FRETMP    =$E604  122.304     122.1132    122.1136    122.1168*   122.1231
   NB        =$E630  122.1183    122.1185    122.1187    122.1192*
   FRETMS    =$E635  121.1392    122.1170    122.1196*
   RTN3      =$E645  122.1197    122.1199    122.1204*
   CHRSTR    =$E646  121.291     122.1206*
   LEFTSTR   =$E65A  121.292     122.1218*
   INS1      =$E660  122.1221*   122.1245
   INS2      =$E666  122.1221    122.1225*
   INS3      =$E667  122.1226*   122.1261
   INS4      =$E668  122.1227*   122.1264    122.1266
M  BPD       =$E67F
   RIGHTSTR=$E686    121.293     122.1241*
   MIDSTR    =$E691  121.294     122.1247*
   INS5      =$E6A2  122.1251    122.1254*
   INSTRNG   =$E6B9  122.1218    122.1241    122.1254    122.1272*
   LEN       =$E6D6  121.287     122.1291*
   GETSTR    =$E6DC  122.1291    122.1293*   122.1299    122.1315
   ASC       =$E6E5  121.290     122.1299*
   GOIQ      =$E6F2  122.1288    122.1300    122.1305*   122.1311    122.1349
   GTBYTC    =$E6F5  121.1439    122.1307*
   GETBYT    =$E6F8  121.1292    122.254     122.1253    122.1308*   122.1345
                     124.25      124.29      124.37      124.43      124.61
                     124.89      124.95      124.102     124.700     124.713
                     124.738     124.742     124.746     124.838
   CONINT    =$E6FB  122.339     122.1206    122.1309*
   VAL       =$E707  121.289     122.1315*
   VL2       =$E70F  122.1316    122.1318*
   VL3       =$E727  122.1326    122.1328*
   POINT     =$E73D  121.1593    122.164     122.1339*
   GTNUM     =$E746  122.1342*   122.1371    122.1377
   COMBYTE   =$E74C  122.1344*   122.1382
   GETADR    =$E752  122.1343    122.1347*   122.1361    124.19      124.129
                     124.143     124.686
   PEEK      =$E764  121.286     122.1357*
   POKE      =$E77B  121.263     122.1371*
   WAIT      =$E784  121.259     122.1377*
   WT2       =$E793  122.1381    122.1383*
   WT3       =$E797  122.1385*   122.1388
   RTN4      =$E79F  122.1389*   123.39
   FADDH     =$E7A0  123.18*     123.781     123.1093
   FSUB      =$E7A7  123.22*     123.210     123.1089    123.1161
```

FSUBT =$E7AA 121.301 123.23* 123.992 123.1086
ADO =$E7B9 123.29* 123.56
FADD =$E7BE 121.1684 123.20 123.31* 123.204 123.216
123.1034 123.1057 123.1075 123.1100
FADDT =$E7C1 121.299 123.27 123.32* 123.696
AD1 =$E7C6 123.32 123.34*
AD2 =$E7CE 123.38* 123.327
AD3 =$E7EA 123.43 123.53*
AD4 =$E7EE 123.52 123.55*
AD5 =$E7FA 123.30 123.42 123.61*
SUBMAN =$E806 123.65 123.67*
SGNIF =$E829 123.75* 123.513 123.598
SIGNIF =$E82E 123.75 123.77* 123.403 123.1068
FLOOP =$E832 123.80* 123.93
ZEROFAC =$E84E 122.1317 123.94* 123.116 123.314
AtoFAC =$E850 123.95* 123.921
AtoFACS =$E852 123.96* 123.304
ADMAN =$E855 123.62 123.99*
FR1 =$E874 123.107* 123.113
FR2 =$E880 123.81 123.113*
FR3 =$E88D 123.105 123.120*
FROUND =$E88F 123.121* 123.488
RTN5 =$E89D 123.120 123.128*
NEGFAC =$E89E 123.76 123.130*
NEG2 =$E8A4 123.131* 123.566
PLUSEPS =$E8C6 123.138* 123.486
RTN6 =$E8D4 123.137 123.139 123.141 123.143 123.145*
OVERFLOW=$E8D5 123.122 123.147* 123.315 123.704
SHFTRES =$E8DA 123.150* 123.244
NXSFT =$E8DC 123.151* 123.162 123.163
SHIFT =$E8F0 123.29 123.161* 123.571
SH1 =$E8FD 123.168* 123.178
SH2 =$E903 123.169 123.171*
SHFTR =$E907 123.60 123.173* 123.581
SH3 =$E911 123.167 123.179*
ONE =$E913 121.1026 121.1027 123.182* 123.208 123.209
123.1150 123.1151
LOGSER =$E918 123.183* 123.211 123.212
SQRhalf =$E92D 123.188* 123.202 123.203
SQRtwo =$E932 123.189* 123.205 123.206
HALFneg =$E937 123.190* 123.214 123.215
LOGtwo =$E93C 123.191* 123.219 123.220
LOG =$E941 121.280 123.193* 123.938
GIQ =$E948 123.194 123.196*
LG2 =$E94B 123.195 123.197*
FMULT =$E97F 123.221* 123.762 123.941 123.964 123.1007
123.1011 123.1025 123.1054
FMULTT =$E982 121.303 123.222*
FMU =$E987 123.222 123.225*
FM1 =$E9B0 123.232 123.234 123.236 123.238 123.243*
FM2 =$E9B5 123.240 123.243 123.246*
FM3 =$E9B8 123.248* 123.262
FM4 =$E9D4 123.249 123.255*
RTN7 =$E9E2 123.223 123.263*
CONUPK =$E9E3 123.22 123.31 123.221 123.267* 123.344
ADEXP =$EA0E 123.225 123.292* 123.351
ADEX2 =$EA10 123.293* 123.1000

ADEX3 =$EA1B 123.296 123.300*
ADEX4 =$EA26 123.303 123.305*
OUTOFRNG=$EA2B 123.309* 123.974
ZERO =$EA31 123.293 123.300 123.312*
JOV =$EA36 123.297 123.311 123.315* 123.324 123.329
123.353
MUL10 =$EA39 123.319* 123.667 123.679 123.775
RTN8 =$EA4F 123.321 123.330*
NUM10 =$EA50 123.332* 123.337 123.338
DIV10 =$EA55 123.336* 123.663 123.778
DIV =$EA5E 123.340* 123.1081
FDIV =$EA66 123.207 123.344* 123.1124 123.1152
FDIVT =$EA69 121.305 123.342 123.345*
FD1 =$EA80 123.356* 123.382
FD2 =$EA96 123.358 123.361 123.364 123.367* 123.381
123.383
FD3 =$EAA3 123.369 123.375* 123.392
FD4 =$EAA6 123.377* 123.390
FD5 =$EAB4 123.376 123.384*
FD6 =$EAD1 123.372 123.391*
FD7 =$EAD5 123.373 123.393*
DIVZ =$EAE1 123.345 123.399*
RES>FAC =$EAE6 123.241 123.398 123.401*
MOVFM =$EAF9 121.1028 121.1678 122.224 123.341 123.408*
123.916 123.1049 123.1117
MOV2F =$EB1E 123.430* 123.1015
MOV1F =$EB21 123.432* 123.1005 123.1108
? MOVML =$EB23 123.433*
SETFOR =$EB27 121.1362 121.1685 123.435*
MOVMF =$EB2B 122.853 123.434 123.437* 123.925 123.1071
MOVFA =$EB53 123.33 123.460*
MFA =$EB55 123.461* 123.935
MFA2 =$EB59 123.463* 123.466
MOVAF =$EB63 123.319 123.336 123.470* 123.689 123.913
123.1077 123.1082
MAF =$EB66 123.471* 123.970
MAF2 =$EB68 123.472* 123.475
RTN9 =$EB71 123.477* 123.483 123.485 123.487
RNDB =$EB72 121.1353 122.111 123.346 123.437 123.470
123.482*
ROUND =$EB7A 123.486* 123.968
SIGN =$EB82 121.1034 123.193 123.490* 123.499 123.526
123.1044
SIGN1 =$EB86 123.492* 123.529
SIGN2 =$EB88 123.493* 123.553
RTN10 =$EB8F 123.491 123.495 123.497*
SGN =$EB90 121.270 123.499*
FLOAT =$EB93 122.337 123.500* 123.691
FLO1 =$EB9B 122.789 123.504*
FLO2 =$EBA0 123.507* 123.731
ABS =$EBAF 121.272 123.515*
FCOMP =$EBB2 122.292 122.512 123.520* 123.767 123.772
123.931
FCOMP2 =$EBB4 121.1687 123.521*
FC1 =$EBE9 123.531 123.535 123.539 123.543 123.550*
FC2 =$EBEF 123.551 123.553*
QINT =$EBF2 122.514 122.1350 123.557* 123.588 123.782

```
QI1        =$EC06  123.562      123.568*
RTN11      =$EC11  123.549      123.573*
QI2        =$EC12  123.570      123.575*
INT        =$EC23  121.271      123.585*     123.928      123.975      123.1083
ZFAC       =$EC40  123.558      123.600*
RTN12      =$EC49  123.587      123.605*
FIN        =$EC4A  121.1600     122.147      122.1335     123.609*
FIN2       =$EC4E  123.611*     123.613
FIN3       =$EC5D  123.616      123.619*
EVAL       =$EC61  123.618      123.621*     123.650      123.684
NXDIGIT    =$EC64  123.614      123.622*
CHKDP      =$EC66  123.620      123.623*
SETSGN     =$EC85  123.630      123.632      123.639*
DPDIG      =$EC87  123.634      123.636      123.640*     123.716
GOGEX      =$EC8A  123.628      123.641*
SGNCHK     =$EC8C  123.637      123.642*
SETDP      =$EC98  123.624      123.648*
ADJEXP     =$EC9E  123.626      123.643      123.657*
AEX        =$ECA0  123.647      123.658*
DPLEFT     =$ECA9  123.663*     123.665
DPRIGHT    =$ECB2  123.662      123.667*     123.669
EVDONE     =$ECB9  123.661      123.666      123.670*
EVD        =$ECBE  123.671      123.673*
INSRTDIG   =$ECC1  123.622      123.675*
NDP        =$ECC8  123.677      123.679*
ADDACC     =$ECD5  123.218      123.683      123.688*
GETEXP     =$ECE8  123.641      123.698*
MVDG       =$ECF7  123.700      123.705*
STEX       =$ED05  123.703      123.715*
HMmiTNTH   =$ED0A  123.718*     123.770      123.771
BILmiONE   =$ED0F  123.719*     123.765      123.766
BILLION    =$ED14  123.720*     123.760      123.761
INPRT      =$ED19  121.577      123.722*
LINPRT     =$ED24  121.925      121.1073     123.727*
?  PRNTFAC =$ED2E  123.732*
PRSTR      =$ED31  123.724      123.733*
FOUT       =$ED34  121.1420     123.732      123.738*
FACSTRNG   =$ED36  122.880      123.743*
SFSG       =$ED41  123.746      123.749*
NOTZE      =$ED4F  123.754      123.756*
MB         =$ED57  123.758      123.760*
STE        =$ED60  123.759      123.764*
CMPBM1     =$ED62  123.765*     123.780
CMPHM      =$ED6D  123.770*     123.777
JM10       =$ED78  123.773      123.775*
JD10       =$ED7F  123.769      123.778*
ROUN       =$ED86  123.774      123.781*
INTPART    =$ED89  123.768      123.782*
DPLOC      =$ED9E  123.787      123.793*
DPL        =$ED9F  123.789      123.794*
PUTDP      =$EDAA  123.798      123.800*
SVY        =$EDBB  123.805      123.809*
MAKSTR     =$EDBD  123.799      123.810*
MSLUP      =$EDC1  123.812*     123.827      123.829      123.856
PARITY?    =$EDE5  123.826      123.829*
COUNTED    =$EDE7  123.828      123.830*
MAKDIGIT   =$EDEE  123.831      123.834*
```

SAVY =$EE09 123.845 123.849*
MVBACK =$EE19 123.858* 123.861
NEEDEX? =$EE26 123.863 123.865*
PUTEX =$EE36 123.868 123.874*
WHATX =$EE42 123.880* 123.882
WNDUP =$EE57 123.755 123.890*
MARKEND =$EE5A 123.867 123.891*
PNTSTK =$EE5F 123.889 123.893*
HALF =$EE64 123.18 123.19 123.897* 123.914 123.915
DECTBL =$EE69 123.814 123.817 123.820 123.823 123.855
123.901*
TEND =$EE8D 123.855 123.911*
SQR =$EE8D 121.278 123.913*
FPWRT =$EE97 121.307 123.918*
PW1 =$EEA0 123.920 123.923*
PW2 =$EEBA 123.927 123.932 123.935*
NEGOP =$EED0 121.313 123.673 123.946* 123.993 123.1097
123.1103 123.1145 123.1164
RTN13 =$EEDA 123.945 123.947 123.949*
LOGe =$EEDB 123.951* 123.962 123.963
EXPSER =$EEE0 123.952* 123.994 123.995
EXP =$EF09 121.281 123.918 123.942 123.962*
X1 =$EF19 123.967 123.969*
OOR =$EF24 123.974* 123.979
X2 =$EF27 123.973 123.975*
X3 =$EF37 123.984* 123.989
ODDSER =$EF5C 123.213 123.1003* 123.1106 123.1155
SERIES =$EF72 123.996 123.1013*
SERMAIN =$EF76 123.1008 123.1015*
SS =$EF85 123.1021 123.1023*
SERLOOP =$EF89 123.1025* 123.1038
NXTERM =$EF96 123.1030 123.1032*
RTN14 =$EFA5 123.1039* 123.1051
RNDADJ1 =$EFA6 123.1041* 123.1052 123.1053
RNDADJ2 =$EFAA 123.1042* 123.1055 123.1056
RND =$EFAE 121.279 123.1044*
RD1 =$EFCC 123.1046 123.1058*
RD2 =$EFE7 123.1071* 123.1114
COS =$EFEA 121.282 123.1073*
SIN =$EFF1 121.283 123.1077* 123.1111
SI1 =$F023 123.1092 123.1097* 123.1127
SI2 =$F026 123.1095 123.1098*
SI3 =$F033 123.1102 123.1104*
TAN =$F03A 121.284 123.1108*
TAN2 =$F062 123.1121 123.1126*
PIhalf =$F066 123.1073 123.1074 123.1129* 123.1159 123.1160
PIdoub =$F06B 123.1078 123.1079 123.1130*
QUARTER =$F070 123.1087 123.1088 123.1098 123.1099 123.1131*
SINSER =$F075 123.1104 123.1105 123.1132*
ATN =$F09E 121.285 123.1142*
ATN1 =$F0A6 123.1144 123.1146*
ATN2 =$F0B4 123.1149 123.1153*
ATN3 =$F0C7 123.1158 123.1162*
RTN15 =$F0CD 123.1163 123.1165*
ATNSER =$F0CE 123.1153 123.1154 123.1167*
ZPSTUFF =$F10B 123.1184* 123.1189 123.1220
RTN16 =$F122 123.1187 123.1194*

```
   COLDST   =$F128  122.365      123.1198*    123.1203     123.1204
   MVZP     =$F152  123.1220*    123.1224
   FNDMEMHI=$F181  123.1243*    123.1252
   MEMFOUND=$F195  123.1248     123.1253*
   CALL     =$F1D5  121.218      124.18*
   INNU     =$F1DE  121.217      124.25*
   PRNU     =$F1E5  121.216      124.29*
   PLOTFNS  =$F1EC  122.227      124.37*      124.52       124.66
   GOERR    =$F206  124.39       124.45       124.50*      124.63       124.70
                    124.77       124.85       124.99       124.711
   LINCOOR  =$F209  124.52*      124.73       124.81
   AT?      =$F218  124.54       124.59*
   PLOT     =$F225  121.219      124.66*
   HLIN     =$F232  121.220      124.73*
   VLIN     =$F241  121.221      124.81*
   COLOR    =$F24F  121.238      124.89*
   VTAB     =$F256  121.240      124.95*
   SPEED    =$F262  121.247      124.102*
   TRACE    =$F26D  121.233      124.110*
   NOTRACE  =$F26F  121.234      124.112*
   NORMAL   =$F273  121.235      123.1209     124.116*
   INVERSE  =$F277  121.236      124.118*
   NRM      =$F279  124.117      124.119*
   SI       =$F27B  124.120*     124.126
   FLASH    =$F280  121.237      124.124*
   HIMEM    =$F286  121.241      124.128*
   JMM      =$F296  124.132*     124.145      124.147
   SETHI    =$F299  124.131      124.134*
   LOMEM    =$F2A6  121.242      124.142*
   ONERR    =$F2CB  121.243      124.151*
   HANDLERR=$F2E9  121.560      121.1115     121.1505     124.162*
   RESUME   =$F318  121.244      124.176*
   JSYN     =$F32E  124.182*     124.184
   DEL      =$F331  121.211      124.184*
   MOVDWN   =$F365  124.197      124.199*
   MOVIT    =$F367  124.200*     124.205
   NDX      =$F388  124.208      124.210*
   GR       =$F390  121.214      124.215*
   TEXT     =$F399  121.215      124.219*
   STORE    =$F39F  121.246      124.225*
   SLI      =$F3AF  124.232      124.234*
   RECALL   =$F3BC  121.245      124.240*
   MOK      =$F3D2  124.248      124.250*
   HGR2     =$F3D8  121.222      124.255*
   HGR      =$F3E2  121.223      124.259*
   SETHPG   =$F3EA  124.258      124.262*
?  HCLR     =$F3F2  124.265*
?  BKGND    =$F3F6  124.267*
   BKGND1   =$F3FE  124.271*     124.275      124.279
   HPOSN    =$F411  124.284*     124.327      124.771
   HPOSN1   =$F441  124.313*     124.315
   HPOSN2   =$F442  124.310      124.314*
   HPLOT0   =$F457  124.327*     124.725
   LFTRT    =$F465  124.337*     124.398      124.497
   LEFT     =$F467  124.338*
   LR1      =$F46E  124.342*     124.362
   LEFT1    =$F471  124.340      124.344*
```

```
   LEFT2    =$F476  124.345     124.347*
   NEWNDX   =$F478  124.348*    124.366     124.368
?  CSHIFT   =$F47C  124.350*
   CSHFT2   =$F47E  124.273     124.324     124.351*
   RTS1     =$F489  124.353     124.357*
   RIGHT    =$F48A  124.337     124.359*
   LRUDX1   =$F49C  124.369*    124.666
   LRUDX2   =$F49D  124.370*    124.670
   LRUD1    =$F4B3  124.381*    124.614
   LRUD2    =$F4B4  124.382*    124.620
   LRUD3    =$F4C4  124.376     124.380     124.388     124.390*
   LRUD4    =$F4C8  124.372     124.384     124.392*
   EQ3      =$F4CD  124.395*    124.406
   UPDOWN   =$F4D3  124.399*    124.516
?  UP       =$F4D5  124.400*
   UP1      =$F4EB  124.407     124.411*
   UP5      =$F4F6  124.415     124.417*
   UP2      =$F4FB  124.405     124.420*
   UP3      =$F4FD  124.410     124.419     124.421*
   UP4      =$F4FF  124.403     124.422*
   UPDWN1   =$F501  124.423*    124.430     124.447
   DOWN     =$F505  124.399     124.426*
   EQ4      =$F508  124.428*    124.435
   DOWN3    =$F524  124.440     124.442*
   DOWN1    =$F52A  124.432     124.445*
   DOWN2    =$F52C  124.436     124.444     124.446*
?  HLINRL   =$F530  124.451*
   HGLIN    =$F53A  124.457*    124.735
   HLIN2    =$F550  124.464     124.471*
   HLIN3    =$F568  124.482     124.485*
   MOVEX    =$F57C  124.496*    124.515
   MOVEX2   =$F581  124.495     124.499*
   HCOUNT   =$F58B  124.504*    124.523
   HLIN4    =$F59E  124.511     124.514*
   MSKTBL   =$F5B2  124.318     124.525*
   EQ1C     =$F5B9  124.402     124.429     124.533*
   COSTBL   =$F5BA  124.538*    124.593     124.597     124.645     124.649
?  HFIND    =$F5CB  124.545*
   HFIND1   =$F5F0  124.569*    124.571
   HFIND2   =$F5FE  124.576     124.578*
   RTS2     =$F600  124.513     124.579*
?  DRAW0    =$F601  124.583*
   DRAW1    =$F605  124.585*    124.776
   DRAW2    =$F626  124.604*    124.627     124.630
   DRAW3    =$F630  124.609*    124.622
   DRAW4    =$F63D  124.613     124.616*
   DRAW5    =$F648  124.619     124.621*
?  XDRAW0   =$F65D  124.635*
   XDRAW1   =$F661  124.637*    124.778
   XDRAW2   =$F682  124.656*    124.677     124.680
   XDRAW3   =$F68C  124.661*    124.672
   XDRAW4   =$F699  124.665     124.668*
   XDRAW5   =$F6A4  124.669     124.671*
M  NI       =$F6B4
   HFNS     =$F6B9  124.685*    124.724     124.730     124.770
   HFNS1    =$F6CD  124.690     124.694*
   GGERR    =$F6E6  124.691     124.693     124.702     124.711*    124.715
```

```
                    124.752
HCOLOR   =$F6E9     121.224     124.713*
RTS3     =$F6F5     124.718*    124.728
COLORTBL=$F6F6      124.716     124.720*
HPLOT    =$F6FE     121.225     124.722*
HP2      =$F708     124.726*    124.736
HP3      =$F70F     124.723     124.729*
ROT      =$F721     121.230     124.738*
SCALE    =$F727     121.231     124.742*
DRWPNT   =$F72D     124.746*    124.775     124.777
DP1      =$F741     124.751     124.753*
DP2      =$F747     124.754     124.757*
DP3      =$F766     124.768     124.772*
DRAW     =$F769     121.226     124.775*
XDRAW    =$F76F     121.227     124.777*
SHLOAD   =$F775     121.232     124.780*
SL1      =$F796     124.798     124.800*
SL2      =$F7A0     124.803     124.805*
SL3      =$F7A3     124.804     124.806*
TAPEPNT  =$F7BC     124.237     124.250     124.819*
GETARYPT=$F7D9      124.225     124.240     124.828*
HTAB     =$F7E7     121.228     124.838*
HTAB1    =$F7EC     124.841*    124.847
HTAB2    =$F7FA     124.842     124.848*
MONPLOT  =$F800     121.196*    124.71
HLINE    =$F819     121.197*    124.79
VLINE    =$F828     121.198*    124.87
SETCOL   =$F864     121.199*    124.91
SCRN     =$F871     121.192*    122.230
PREAD    =$FB1E     121.193*    122.340
SETTXT   =$FB39     121.202*    124.220
SETGR    =$FB40     121.201*    124.217
TABV     =$FB5B     121.200*    124.100
HOME     =$FC58     121.185*    121.229
MONWAIT  =$FCA8     121.191*    121.1487
RD2BIT   =$FCFA     121.186*    124.815
RDKEY    =$FD0C     121.188*    121.702     121.1557
GETLN    =$FD6A     121.187*    121.684
COUT     =$FDED     121.190*    121.1483
INPORT   =$FE8B     121.194*    124.27
OUTPORT  =$FE95     121.195*    124.31
WRITE    =$FECD     121.189*    121.1151    121.1153    124.236     124.238
MONREAD  =$FEFD     121.203*    121.1155    121.1159    124.241     124.251
                    124.787
MONREAD2=$FF02      121.204*    124.817
```